

情報系学科達成度調査結果報告

1. はじめに

今回の達成度調査は、情報および情報関連分野の専門学科での教育・学習によって、卒業生がどのような知識・能力をもつことになるのかについて、現状を把握することが目的であった。

そもそも、4年間の教育・学習によって学生に達成させようとする内容・水準の設定は、学科・学部・大学が独自に行うものであり、それぞれ異なっていて不思議はない。特に、情報および情報関連分野は、他分野に比してその広がりが大きく、代表的なCS (Computer Science), CE (Computer Engineering), IS (Information Systems), SE (Software Engineering) の4領域にとどまらず、多様な領域を内包しているため、こうした設定には大きな差異がある。その差異を認めたとしても、なお、この分野の学科にあっては何らかの知識・能力を獲得させるはずの、“共通する”項目について、その達成度がどのようなものになっているかを調べたのである。

達成度を調査する対象項目の選定に当たっては、日本技術者認定機構(JABEE) (<http://www.jabee.org/>)の認定基準(pdfファイル)を参考とした。認定基準 ([http://www.jabee.org/OpenHomePage/criteria2004-6\(2005.11.11\).pdf](http://www.jabee.org/OpenHomePage/criteria2004-6(2005.11.11).pdf)) は、教育プログラム(学科・専修・コースなど)がつぎの(a)-(h)に関して学習・教育の目標を設定するように求めている。

- (a) 多面的に考える能力・素養
- (b) 技術者倫理
- (c) 数学・自然科学・情報技術の知識・応用能力
- (d) 専門分野の知識・応用能力
- (e) デザイン能力
- (f) コミュニケーション基礎能力
- (g) 継続的学習能力
- (h) 与えられた制約の下で計画的に仕事を進め、まとめる能力

この(d)に関しては、それぞれの専門分野に応じて分野別要件が認定基準の補則として設けられている。情報および情報関連分野の分野別要件 (<http://www.ipsj.or.jp/12kyoiku/acre/criteria-info.html>) は、つぎのように定めている。

修得すべき知識・能力

教育プログラムの修了生は、つぎに示す知識・能力を身に付けている必要がある。

1. つぎの学習域すべてにわたる、理論から問題分析・設計までの基礎的な知識およびその応用能力
 - i. アルゴリズムとデータ構造
 - ii. コンピュータシステムの構成とアーキテクチャ
 - iii. 情報ネットワーク
 - iv. ソフトウェアの設計
 - v. プログラミング言語の諸概念
2. プログラミング能力
3. 離散数学および確率・統計を含めた数学の知識およびその応用能力
4. 教育プログラムが対象とする領域に固有の知識およびその応用能力

このうち4は、そのプログラムがCS、CE、IS、SEなどのどの領域を対象としているかによって変わってくる。2005年度では、この分野別要件に示された1～3の項目を主対象として調査を行い、2006年度にはCS、CE、IS、SEの領域を含めて調査を行なった。

2. 2005年度実施内容

理工系情報学科協議会(<http://kyougikai.meme.hokudai.ac.jp/>)に参加している学科を対象として、2005年7月3日～7月30日に調査を行った。各学科には、説明文・調査票をシートとしてまとめたExcelファイルを添えた依頼状をメールで送付した。

調査対象は、分野別要件を参考に、8個の大項目(20の小項目)とした(表1)。達成度そのものは、0～5の達成度区分を記入してもらうとともに、その達成度を具体的に文章の形でも記入してもらった(表2)。

達成度は、それぞれの学生の学習状況によって変わってくる。必修科目で学習・教育が行われている項目なら、学生によらずこれだけは達成していると言う意味での達成度を示す事ができる。しかし、選択科目によっている項目に対してはそれを示すのが難しい。そこで、その学科での平均レベルの学生が達成していると思っていい達成度と、最低レベルの学生でも達成しているはずの達成度とを別々に答えてもらった。

3. 2005 年度調査結果

回答があったのは、28学科であった。寄せられた回答には、書かれた具体的な文章から読み取れる達成度と、記入された達成度区分とにそごがみられるものがあった。回答にあった達成度そのものと、具体的な文章から読み取れる達成度に整理したものの二通りの集計結果を示す(表3,図1)。

“回答”は、回答にあったものの集計を示し、“判定”は、整理後の集計を示す。なお、項目によって回答がなかったものがあった。また、回答された達成度区分が“2～4”などと幅を持たせた形になっていたものは“回答”には計上していない。

この集計結果から何がいえるかについての議論はあえてしないでおく。ただ、“2～4”といった達成度区分の記入があったことからわかるように、回答にあたって、達成度を達成度区分で表すことの難しさ、それを具体的に文章で書き下すことの難しさを感じた方が多かったようである。また、設問でいう“平均的学生”をどのように想定すればよいかに戸惑われた方もあったと思われる。こうした点を改善して、より多くの学科からの回答が得られるようにして再度調査を行う予定でいる。

4. 2006 年度実施内容

2005年度の調査において、項目の解釈の違いや達成度の考え方の違いが顕著に現れる場合があったことから、各学習域に対象とする分野等の説明を増やし、見本となる達成度を公開することで、回答をしやすく、かつばらつきが少なくなるように試みた。2006年度は2005年度と同じく理工系情報学科協議会に参加している学科に電子メールで調査を依頼した。2005年に回答のあった学科については、達成度を点検した結果を電子メールで通知し、再度の協力を依頼した。なお、調査には、達成度調査の領域(表4)、達成度区分(表5)、達成度記事術例(表6)を示して行った。回答はExcelファイルによる電子メールでも、ウェブページからでも回答できるようにした。調査期間は2006年12月～2007年2月まで行なった。

アルゴリズムとデータ構造、コンピュータシステムの構成とアーキテクチャ、情報ネットワーク、ソフトウェアの設計、プログラミング言語の諸概念、プログラミング能力、離散数学および確率・統計を含めた数学の知識およびその応用能力については、先に延べたように達成度の見本を示し、2005年度の回答で分野の領域の解釈に違いが多いものについては補足説明をした。見本は2005年度の回答から適切と思われるものを中心に選んだ。特定の記載方法に片寄らないように、また各達成度区分で想定される範囲も考慮の上選出した。

新規に追加したCS, CE, SE, ISの4領域については、達成度の見本を示さなかった。CS, CE, SE, ISの領域には重なりあう分野は類似の項目もあるが、それぞれで単独の調査が行なえるように、別個の項目として調査した。

5. 2006 年度調査結果

回答があったのは26学科であった。そのうち、13学科は2005年度に引き続いて回答があり、13学科は新規の回答であった。一部の学科は達成度区分のみの記載で、具体的な達成度の記述がない。これは2005年度でも同様である。

2006年の回答結果と2005年の回答結果を比較可能なようにまとめた表(表7)とグラフ(図2)に示した。2006年の達成度区分の集計結果は、達成度区分の調整をしていない。なお、2006年は調整を行っていないため、個々の回答には達成度の内容と区分とにずれが見られるものがあるが、全体の傾向を知ることができる。

CSとCEの分野の回答はかなりあった。CSとCEでは共通分野との違いが明確でないものも散見され、項目の設定も含めて検討が必要である。SEとISは達成度としての記述がかなり難しい項目があり(たとえばコミュニケーション能力)、また実際に該当する学科等は少ないと思われたが、回答もわずかであった。しかし、かなり詳細な回答もある。

6. 今後の計画

2006年度の達成度調査の目的として、理工系情報学科協議会会員学科以外に、広く情報系学科に対して調査することもあったが、回答にかなりの労力を要することや、学部・学科再編等が活発であるため、どのような調査をするかという問題がある。この点を踏まえ、調査方法を見直し、情報系学科全般に渡る調査を計画している。

表1. 大項目小項目小項目の内容

この表は大項目および小項目の内容を分かりやすく示すために挙げたキーワード例であり、これらの内容の全てを教育しなければならぬことを示すものではありません。

大項目	小項目	小項目の内容
アルゴリズムとデータ構造	基本データ構造	配列, 文字列, スタック, リスト
	基本探索	線形探索, 二分探索, ハッシュ法, 二分探索木
	基本アルゴリズム	整列(ソート), 文字列照合, グラフアルゴリズム
	アルゴリズムの効率	計算量, 0記法
	数とデータの表現	2進数, 浮動小数点数, 文字コード
	情報・メディアの表現	デジタル表現, 標本化・量子化, データ圧縮, 品質
	コンピュータシステムの構成とアーキテクチャ	組み合わせ論理回路, 順序回路, 命令セット, アドレッシング, 割り込み, メモリ階層, キャッシュメモリ, 並列処理
	オペレーティングシステム	プロセス, マルチプログラミング, スケジューリング, ファイルシステム, 逐次アクセス, ランダムアクセス, ディレクトリ, 仮想記憶, キャッシュメモリ, ページング
	コンピュータネットワーク	パケット交換, OSI 参照モデル, 通信プロトコル, ローカルエリアネットワーク
	インターネット	IP アドレス, 名前解決, 経路制御, 電子メール, WWW, ファイル転送, サーバエンジン
情報ネットワーク	ネットワークセキュリティ	ファイアウォール, 暗号, 認証
	ソフトウェアプロセス	ライフサイクル, ウォーターフォールモデル, スパイラルモデル
	ソフトウェアの設計技法	モジュール設計, コンポーネント, 再利用, CASE ツール, オブジェクト指向
	データの設計と管理	データベース, 問い合わせ言語, SQL, DBMS, Web 文書, XML
プログラミング言語の諸概念	プログラミング言語の基本概念	変数, データ型, 制御構造, サブルーチン, パラメータ, 再帰, モジュール化, オブジェクト指向, 並行プログラミング
	言語処理系の構成	コンパイラ, インタプリタ, オプティマイザ, BNF, プログラミング言語の定義, コンパイラ・コンパイラ, 仮想マシン, ガベージコレクション
	仕様の明確化	要求分析, 仕様設計
プログラミング能力	プログラムの実装	構造化プログラミング, イベント駆動プログラミング, オブジェクト指向プログラミング, スクリプティング
	プログラムの検証・テスト	検証, テスト, デバッグ, 運用管理
	プログラミング環境	エディタ, コンパイラ, デバッガ, 統合開発環境
離散数学		論理学, ブール代数, 集合論, 数え上げ, グラフ理論, 形式言語, 正規表現, BNF
	確率・統計	平均, 分散, 確率分布, 大数の法則, データ解析

表2.学生の達成度に関する具体的記述例

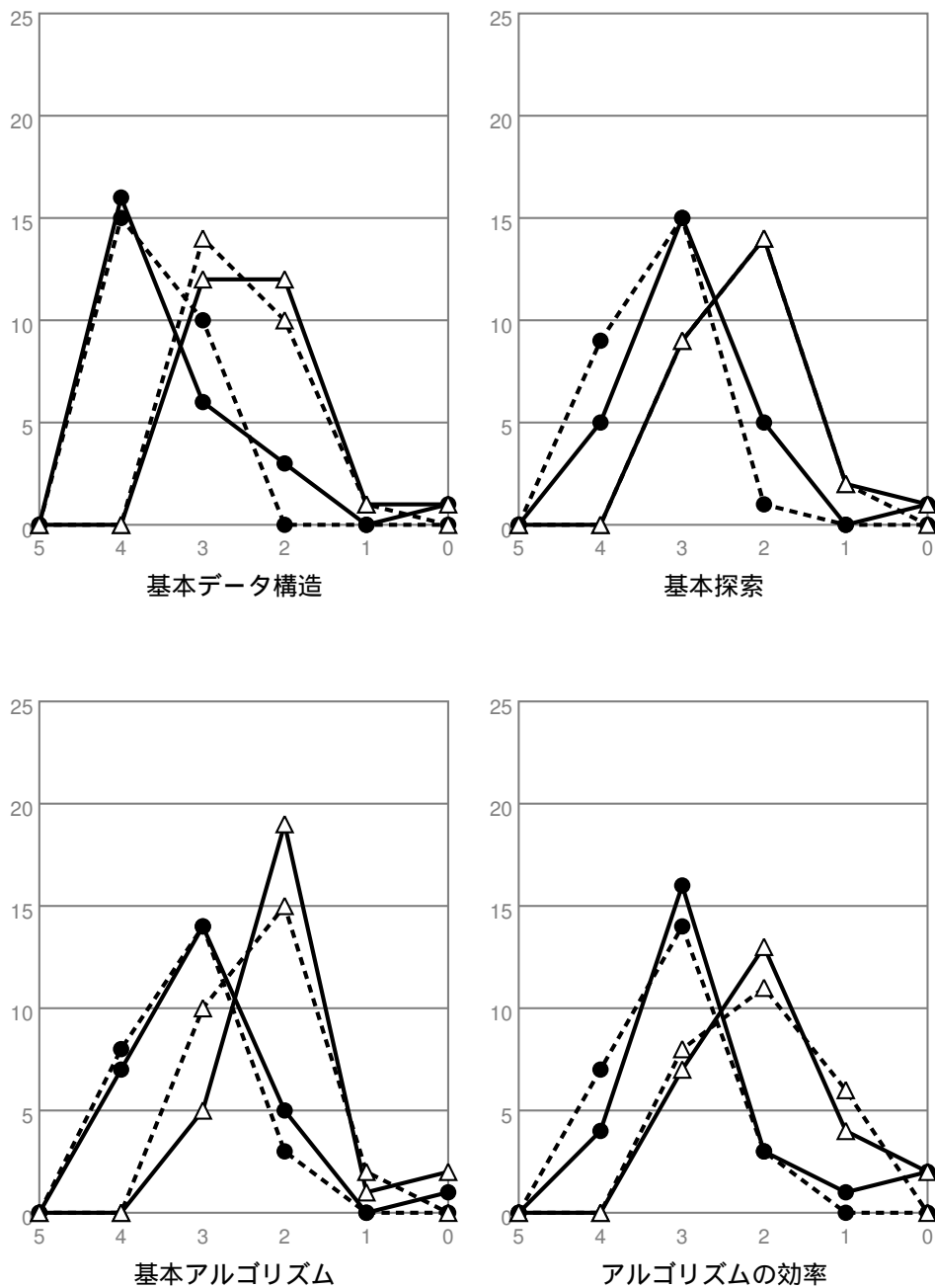
達成度区分	複素数	基本データ構造	ソフトウェアプロセス	仕様の明確化	プログラムの実装	プログラムの検証・テスト	プログラミング環境
0	「複素数って何？」 複素数という用語を知っている	具体的なデータ構造の用語を知っている。 例えば、スタックや配列などの基本データ構造に関する用語を知っている。	具体的なソフトウェアプロセスの用語を知っている。 例えば、ウォータフォールモデルやスパイラルモデルなどのソフトウェアプロセスに関する用語を知っている。	要求、仕様などの用語について知っている。	挨拶文を表示するプログラムが書ける。	検証、テストが何を行うことなのかを知っている。	インタープリタ、またはコンパイラを使うことができる。
1	複素数の概念(実数部と虚数部から構成)を説明できる	具体的なデータ構造を説明できる。 例えば、スタックや配列などについて説明できる。	具体的なソフトウェアプロセスに関する知識を説明できる。 例えば、ウォータフォールモデルについて説明できる。	仕様の意義について説明できる。	データ構造を使用したループ、または再帰を用いたプログラムの概要を説明できる。 プログラムの挙動をトレースできる。	間違いの箇所を指摘されたとき、その箇所がどうして間違っているかを説明できる。	プログラミング環境の機能を説明できる。
2	乗除算ができる	与えられた情報を保持できる配列を定義できる。	指示に従って、ウォータフォールモデルで簡単なプログラミング課題を遂行できる。	サブルーチンの機能が正しく記述できる。 仕様からプログラムの挙動を図示できる。	与えられた仕様に基づき、サブルーチンを作成できる。	仕様通りにプログラムが作成されたかどうかを調べ、誤りを修正できる。	与えられたプログラミング環境を構成できる。
3	インピーダンスの計算に応用できる(CEの場合)	与えられた情報を保持するため、複数のデータ構造の中から適切なデータ構造が選択できる。	与えられた課題に対して、ウォータフォールモデルに沿って作業手順を構成実行できる。	与えられた課題に対して、適切なソフトウェアプロセスを適用し、効果的な作業手順を構成実行できる。			
4	回路解析が自由にできる(CEの場合)	目的に応じて、複数のデータ構造を適切に組み合わせることができる。					
5							

表3. 達成度調査：2005年夏集計結果

大項目	小項目		平均レベルの学生の達成度					最低レベルの学生の達成度						
			5	4	3	2	1	0	5	4	3	2	1	0
アルゴリズムとデータ構造	基本データ構造	回答	0	15	10	0	0	0	0	0	14	10	1	0
		判定	0	16	6	3	0	1	0	0	12	12	1	1
	基本探索	回答	0	9	15	1	0	0	0	0	9	14	2	0
		判定	0	5	15	5	0	1	0	0	9	14	2	1
	基本アルゴリズム	回答	0	8	14	3	0	0	0	0	10	15	2	0
		判定	0	7	14	5	0	1	0	0	5	19	1	2
アルゴリズムの効率	回答	0	7	14	3	0	0	0	0	8	11	6	0	
	判定	0	4	16	3	1	2	0	0	7	13	4	2	
コンピュータシステムの構成とアーキテクチャ	数とデータの表現	回答	0	13	10	1	1	0	0	1	10	12	2	0
		判定	0	6	16	3	1	0	0	0	4	19	3	0
	情報・メディアの表現	回答	0	4	12	5	1	0	0	0	5	12	3	1
		判定	0	5	8	7	1	1	0	0	3	15	2	1
	プロセッサ・アーキテクチャ	回答	0	8	15	2	0	0	0	0	9	14	2	0
		判定	0	8	12	6	0	0	0	0	6	18	2	0
オペレーティングシステム	回答	0	6	12	7	0	0	0	0	5	14	6	0	
	判定	0	2	11	14	0	0	0	0	2	21	4	0	
情報ネットワーク	コンピュータネットワーク	回答	0	5	13	8	0	0	0	0	5	16	4	0
		判定	0	5	7	14	0	0	0	0	4	20	1	0
	インターネット	回答	0	8	12	5	0	0	0	0	9	12	4	0
		判定	0	3	14	8	0	0	0	0	8	17	0	0
	ネットワークセキュリティ	回答	0	0	12	11	2	0	0	0	1	13	10	0
		判定	0	2	6	14	1	2	0	0	0	19	4	1
ソフトウェアの設計	ソフトウェアプロセス	回答	0	5	10	9	2	0	0	0	4	11	8	1
		判定	0	3	11	8	3	1	0	0	3	12	9	1
	ソフトウェアの設計技法	回答	0	6	18	2	1	0	0	0	5	15	6	0
		判定	0	5	15	4	1	2	0	0	5	14	6	1
	データの設計と管理	回答	0	6	14	6	1	0	0	0	7	11	8	0
		判定	0	3	16	5	1	2	0	0	9	7	8	2
プログラミング言語の諸概念	プログラミング言語の基本概念	回答	0	9	15	3	0	0	0	0	10	11	3	0
		判定	0	7	13	5	0	2	0	0	10	12	2	1
	言語処理系の構成	回答	2	6	12	6	0	0	0	1	7	12	5	0
		判定	0	2	9	15	0	1	0	0	3	17	5	1
プログラミング能力	仕様の明確化	回答	0	5	13	6	1	0	0	0	7	11	5	1
		判定	0	5	12	7	1	1	0	0	7	13	3	2
	プログラムの実装	回答	0	14	8	3	0	0	0	0	12	8	3	1
		判定	0	13	11	1	1	0	0	0	17	6	1	1
	プログラムの検証・テスト	回答	0	9	15	2	0	0	0	0	13	8	4	0
		判定	0	9	15	1	0	2	0	0	12	8	5	1
プログラミング環境	回答	1	12	12	0	1	0	0	1	17	3	4	0	
	判定	0	2	21	1	0	3	0	0	18	5	1	2	
数学	離散数学	回答	0	6	18	2	0	0	0	0	6	17	2	0
		判定	0	3	20	4	0	0	0	0	11	13	2	0
	確率・統計	回答	0	2	20	2	0	0	0	0	4	15	3	0
		判定	0	2	20	3	0	1	0	1	8	13	1	1

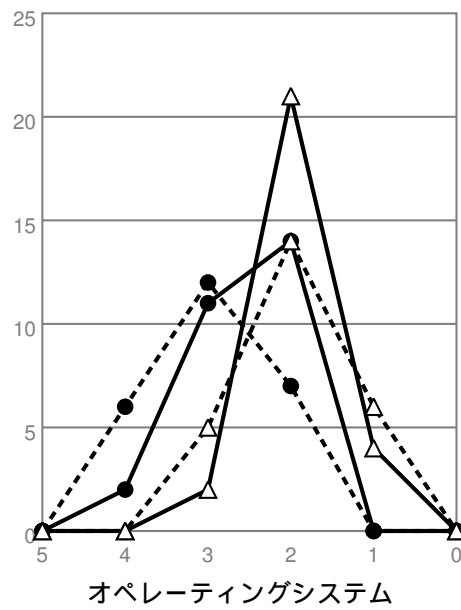
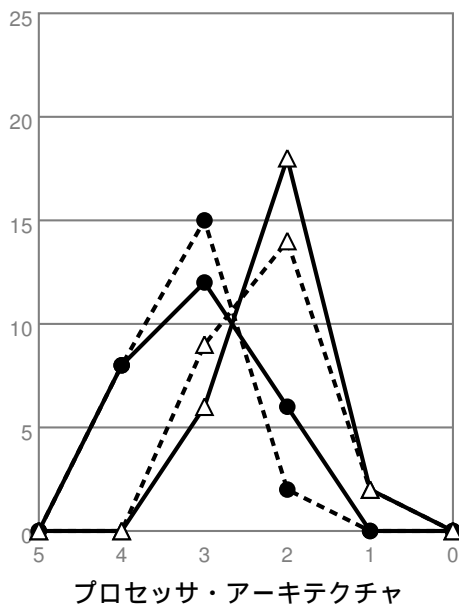
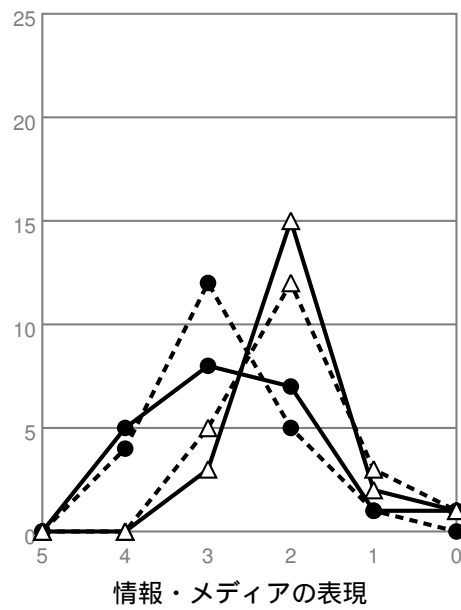
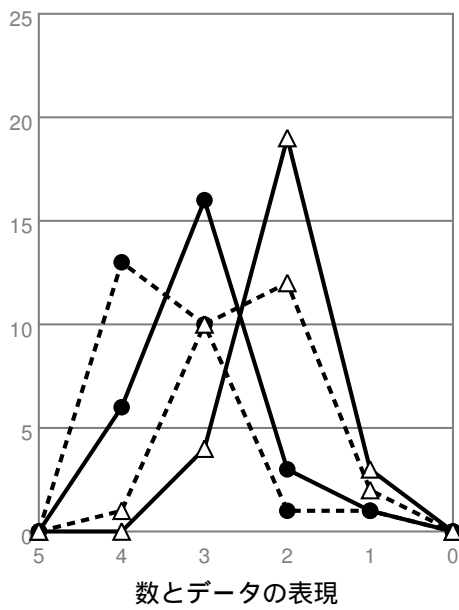
項目によって未回答や集計できない回答があるため、小項目によって回答総数が異なる。

図1：2005年度の達成度区分のグラフ
アルゴリズムとデータ構造



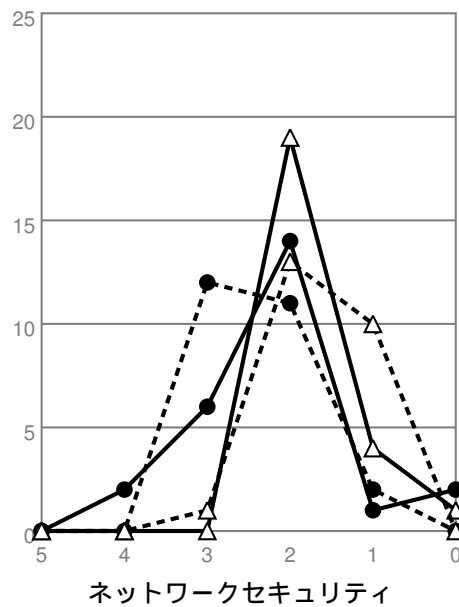
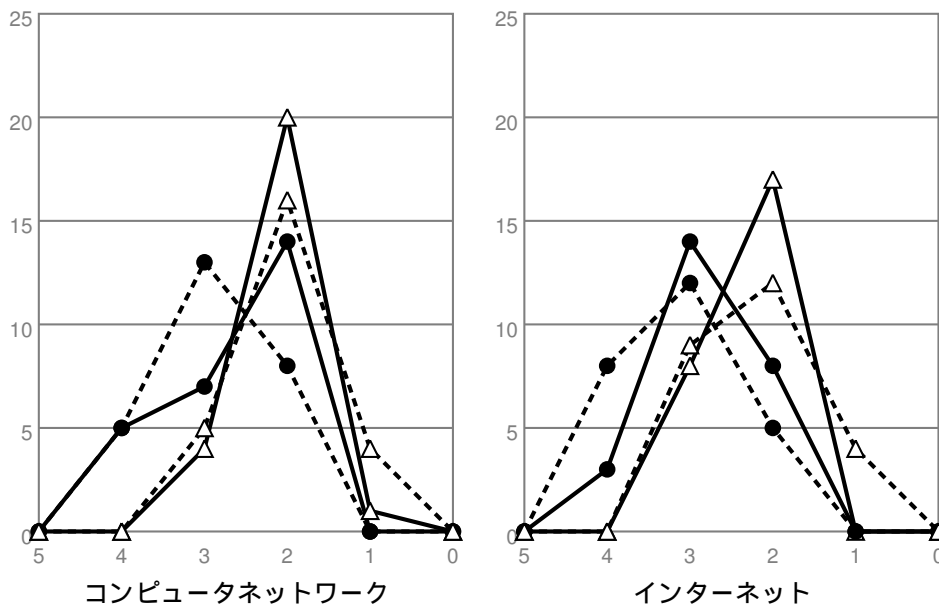
破線は回答そのままの区分、実線は判定後の区分、 \triangle は平均レベル、 \bullet は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

コンピュータシステムの構成とアーキテクチャ



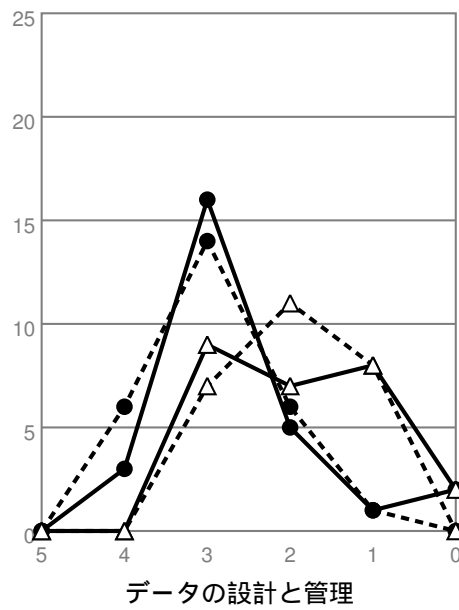
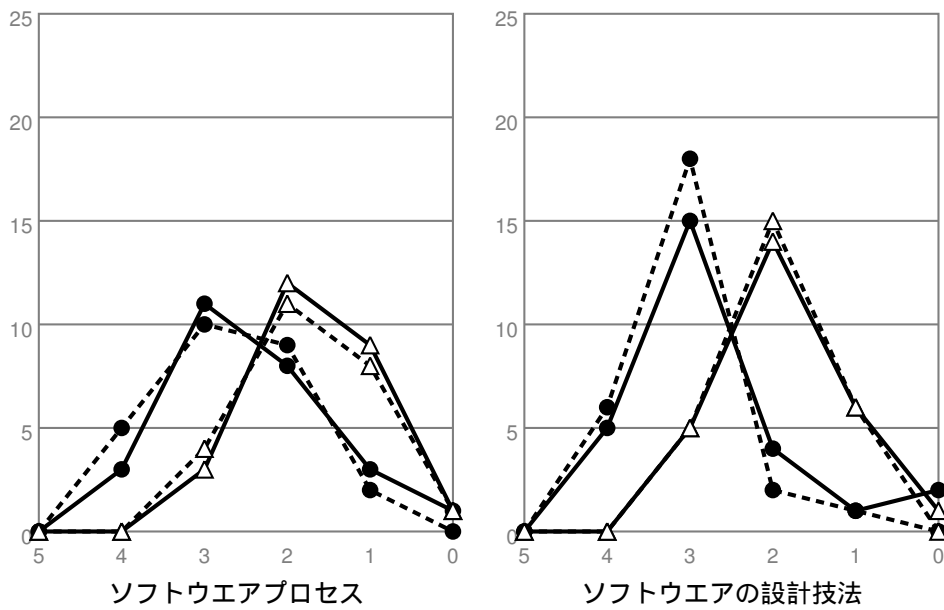
破線は回答そのまの区分，実線は判定後の区分， Δ は平均レベル， \bullet は最低レベルを表す。
縦軸は学科数，横軸は達成度区分を表す。

情報ネットワーク



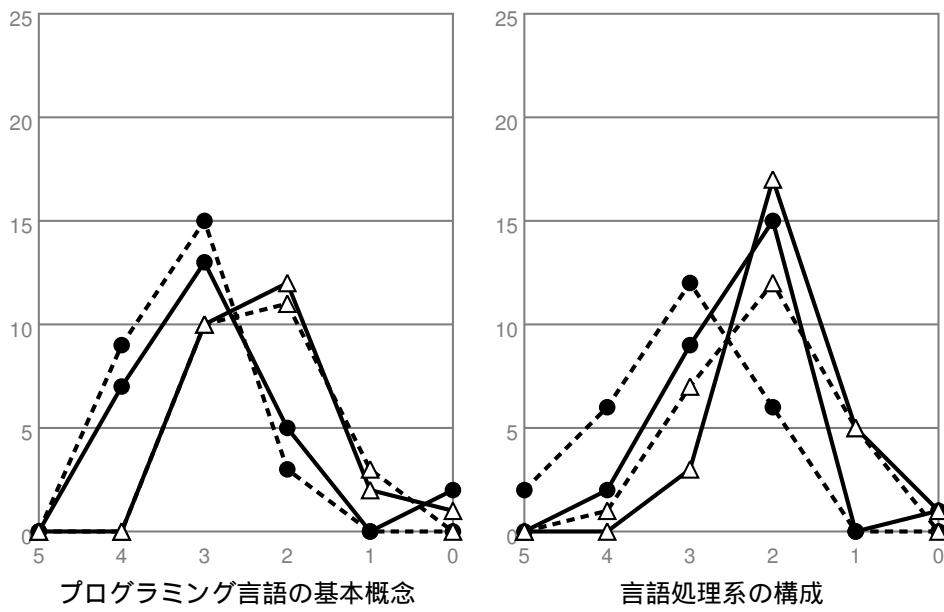
破線は回答そのままの区分，実線は判定後の区分，△は平均レベル，●は最低レベルを表す。
縦軸は学科数，横軸は達成度区分を表す。

ソフトウェアの設計



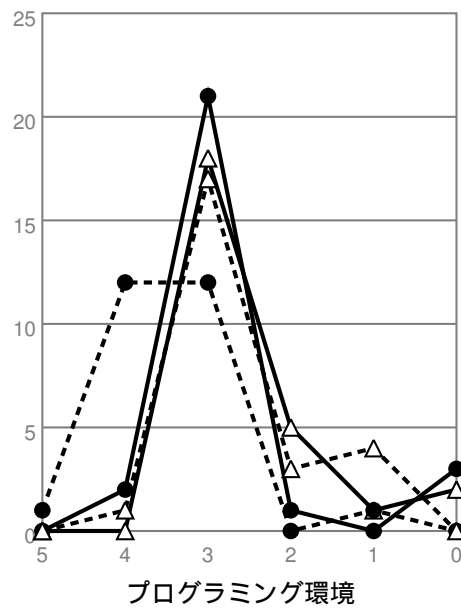
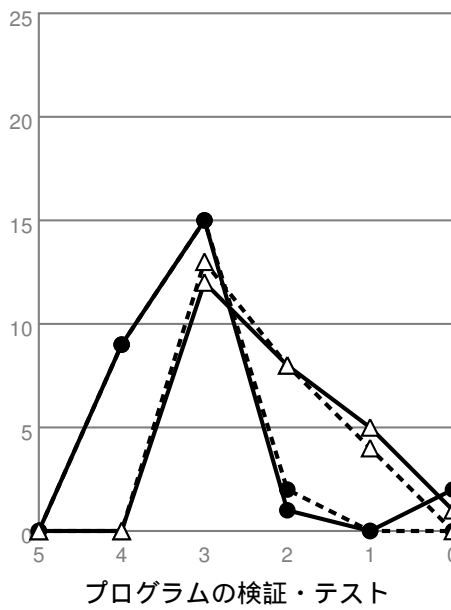
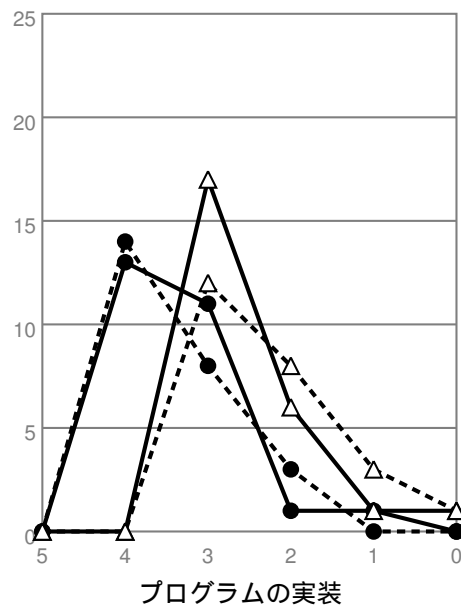
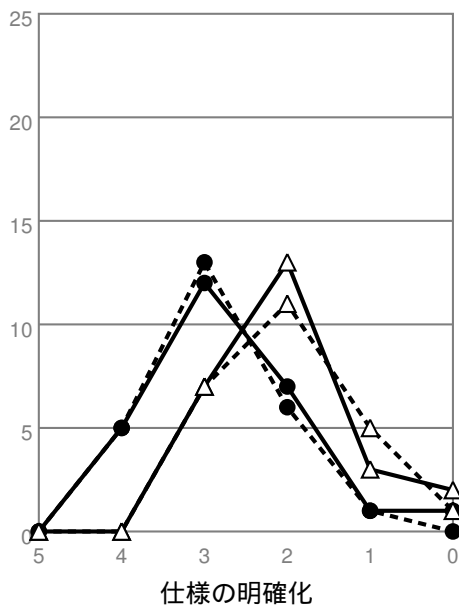
破線は回答そのままの区分，実線は判定後の区分， Δ は平均レベル，● は最低レベルを表す。
縦軸は学科数，横軸は達成度区分を表す。

プログラミング言語の諸概念



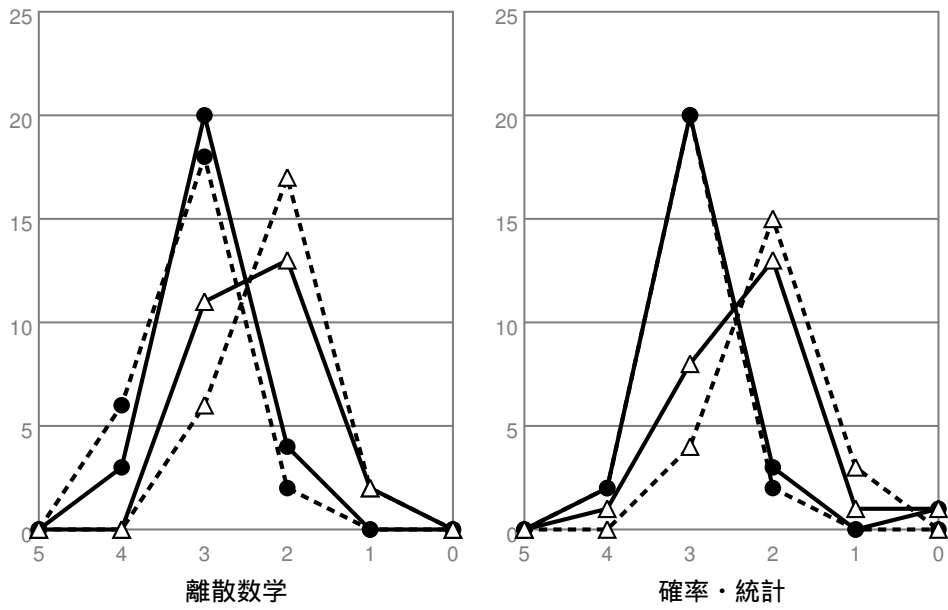
破線は回答そのままの区分，実線は判定後の区分， Δ は平均レベル， \bullet は最低レベルを表す。
縦軸は学科数，横軸は達成度区分を表す。

プログラミング能力



破線は回答そのまの区分，実線は判定後の区分，△は平均レベル，●は最低レベルを表す。
縦軸は学科数，横軸は達成度区分を表す。

数学



破線は回答そのままの区分、実線は判定後の区分、△は平均レベル、○は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

表 4. 2006 年度 達成 度 調査 領域 一 覧

この表は大項目および小項目の内容を分かりやすく示すために挙げたキーワード例でありこれらの内容の全てを教育しなければならないことを示すものではありません。

大項目	小項目	小項目の内容 (例)	補足
アルゴリズムとデータ構造	基本データ構造	配列, 文字列, スタック, リスト	
	基本探索	線形探索, 二分探索, ハッシュ法, 二分探索木	
	基本アルゴリズム	整列 (ソート), 文字列照合, グラフアルゴリズム	
	アルゴリズムの効率	0 記法, $O(1)$, $O(n)$, $O(n^2)$, $O(n \log n)$, 計算量	
コンピュータシステムの構成とアーキテクチャ	数とデータの表現	2 進数, 浮動小数点数, 文字コード	
	情報・メディアの表現	ディジタル表現, 標本化・量子化, データ圧縮, 品質	
	プロセッサ・アーキテクチャ	組合せ論回路, 順序回路, 命令セット, アドレッシング, 割込み, キャッシュメモリ, 多重処理	
	オペレーティングシステム	プロセス, スレッド, スケジューリング, ファイルシステム, 仮想記憶	一般的な利用に関する知識は含まない
情報ネットワーク	コンピュータネットワーク	パケット交換, OSI 参照モデル, 通信プロトコル, ローカルエリアネットワーク	OSI 第 2 層あたりまでを範囲とする
	インターネット	IP アドレス, 名前解決, 経路制御, 電子メール, WWW, ファイル転送, 遠隔ログイン	インターネットの単純な利用に関する知識や Web ページの作成は含まない
	ネットワークセキュリティ	ファイアウォール, 暗号, 認証, ウイルス	
ソフトウェアの設計	ソフトウェアプロセス	ライフサイクル, ウォーターフォールモデル, スパイラルモデル	
	ソフトウェアの設計技法	モジュール設計, コンポーネント, 再利用, オブジェクト指向, 状態遷移図	プログラムの作成そのものは含まない
	データの設計と管理	データベース, SQL, DBMS, XML	
	プログラミング言語の基本概念	変数, データ型, 制御構造, サブルーチン, パラメータ, 再帰, モジュール化, オブジェクト指向, 並行プログラミング	
プログラミング能力	言語処理系の構成	コンパイラ, インタプリタ, 正規表現, BNF, 仮想マシン, ガベージコレクション	プログラムの作成やコンパイラの使用は含まない
	仕様の明確化	要求分析, 仕様設計, ドキュメンテーション	
	プログラムの実装	構造化プログラミング, オブジェクト指向プログラミング, イベント駆動プログラミング, スクリプティング	仕様あるいはアルゴリズムが与えられていることを前提とする
	プログラムの検証・テスト	検証, デバッグ, 単体テスト, 統合テスト	
プログラミング環境	エディタ, コンパイラ, デバッガ, 統合開発環境	プログラミング開発のための利用技術	

大項目	小項目	小項目の内容 (例)	補足
数学	離散数学 確率・統計	集合, 写像, 論理, 数え上げ, グラフ 平均, 分散, 確率分布, 仮説検定, 最小2乗近似	アルゴリズムに関する内容は含まない
Computer Science 専門領域	アルゴリズムとデータ構造 オペレーティングシステム 計算理論		
	コンピュータシステムの構成とアーキテクチャ 情報ネットワーク		
	数値計算・記号計算		
	ソフトウェア方法論・ソフトウェア工学		
	知的システム	人工知能, 知識工学等を含む	
	データベース・情報検索		
	ヒューマンコンピュータインタラクション		
	プログラミング言語 原理の異なる複数のプログラミング言語	プログラミング言語の原理と実装 複数のプログラミング言語でプログラム作成ができる能力	
Computer Engineering 専門領域	システムプログラムに関する基礎的項目		
	コンピュータシステム 情報通信	論理設計, 集積回路, コンピュータアーキテクチャ, 電子回路, デジタル回路, など 信号処理, 符号理論, 情報ネットワーク, デジタル通信, など 画像処理, 音声処理, マルチメディア処理, データベース, 人工知能, など	
	コンピュータ応用		

大項目	小項目	小項目の内容 (例)	補足
Software Engineering 専門領域	情報倫理・社会・法律・経済・安全	ソフトウェア技術者としての社会的責任の遂行と実践に必要な る、情報倫理・社会・法律・経済・安全に関する事項	
	要求分析および設計	ソフトウェアシステムに関わる、要求分析および設計に関する基礎 的技術	
	検証・正当性確認	ソフトウェアシステムに関わる、検証・正当性確認に関する基礎的 技術	
	実現、保守	ソフトウェアシステムに関わる、実現および保守に関する基礎的技 術	
	プロジェクト管理等	ソフトウェアシステム開発の実践に必要な、プロジェクト管理・プ ロダクト構成管理・プロセス管理・リスク管理・品質計量尺度に関 する基礎的技術	
	コミュニケーション能力	ソフトウェア開発プロジェクトに参加するために必要な、つぎの視 点からのコミュニケーション能力(1)提案作成、プレゼンテーショ ン、聞き取りと分析(2)計画、折衝、協調、技術・経済の両面から の意思決定、統括	
	データ管理		
	分析と設計		
	組織における情報システム の役割		
	情報システムを囲む環境		
多様な情報システムの事例 理解			
問題形成・モデリング・プ ロジェクト管理	情報システム開発の実践に必要な問題形成・モデリング・プロジェ クト管理についての十分な量の実習経験を含む		
コミュニケーション能力・ プレゼンテーション能力	コミュニケーション能力・プレゼンテーション能力・深さ 能力を得させるための十分な量と深さ		

表 5. 2006 年度情報系学科 達成度区分

達成度の記入は、学生の達成度に関する具体的記述と、次に挙げる達成度レベルを記載してください。具体的記述は、小項目の領域に関係する具体的内容(たとえば二分探索、順序回路、OSI 参照モデル)と、それに関して、「活用できる」、「使用できる」、「説明できる」といったレベルがわかるように記載してください。

具体的記述に合わせて、達成度を次の 6 段階で記入してください。

達成度区分	達成度の表示例	補足説明	備考
0	聞いたことがない	その項目について学生は何も説明できない。	
1	…を知っている …の知識を持つ	該当項目について講義で聞いていて、その概要を述べることができる。	
2	…を説明できる …を理解している	該当項目を知識として理解しており、その知識を説明できる。または、具体例を挙げることができる。	項目によっては直接に関係する講義がなくても、学生の大多数がこのレベルに達していることがある。
3	…を使用できる …を実行できる	概念と使い方がわかるレベル。限られた条件下(演習課題等)で使っている。または、具体的な指示を与えられれば、該当項目を実行できる。	設計法を実行できる。 証明ができる。 プログラムが説明できる、回路を説明できる、などは区分 3 とすることもできる。授業中の演習や数日～1週間程度かかる問題などができる。
4	…を活用できる …を…に活用できる	詳しく理解し応用できるレベル。現実に近い状況(卒業研究等)で応用できる。大まかな方向性を与えられれば、該当項目を実行できる。	現実に近いレベルで応用できる。 1ヶ月程度以上かかるような演習の課題ができる。 なるべく応用できる範囲を明記する。 項目によっては区分4は、平均的學生ではほとんど達成できないと考えている。
5	…に熟練している	実務(現実社会の問題解決)に応用している。	学部卒業時にこのレベルに達するのはごく一部の學生に限られると考えている。
参考		「小項目のごく一部について使用できる」程度の場合は、区分 3 ではなく区分 2 以下とする。 小項目が含む領域がある程度の広さで「使用できる」や「活用できる」ときに区分 3 や区分 4 とする。 単に「応用できる」といっただけの記載では、どの程度応用できるか判断できないので、区分 3 とする。 単に「卒論に応用できる」というような記載だけの項目では、他に記載されている内容等から区分 3 と解釈されることがある。	

表 6. 情報系学科 達成度調査の記述例

大項目	小項目	達成度区分	学生の達成度に関する具体的記述
アルゴリズムとデータ構造	基本データ構造	4	配列、文字列、スタック、リストについて、その概念と特徴を十分に理解しており、卒業研究等でプログラムを作成する際に、目的に沿った適当な選択を行い、利用することができる。
		3	基本データ構造について説明ができる。指示のもとに構造を利用したプログラムを作成することができる。
		3	スタックなどの基本データ構造を理解し、それを配列等を使ってプログラムとして実現できる。
		2	配列、文字列、スタック、リスト等の基本データ構造を理解し、説明できる。
		1	変数、配列、リスト、スタック、キュー、二分木について知っている。
		3	線形探索、二分探索の方法と特徴を説明できる。B木の意義と特徴を説明できる。ハッシュ法、二分探索木について、特徴を説明し、演習課題において適切に利用できる。
		3	与えられた条件下で適切な探索技法を選択、使用できる。
		2	・線形探索や二分探索の原理を説明できる。 ・ハッシュの概念とチェイン法や開番地法などの特徴、基本操作を説明できる。 ・二分探索木の定義および基本操作(挿入、探索、削除)を説明できる。
		2	線形探索、二分探索、ハッシュ法、二分探索木等の基本探索アルゴリズムを理解し、説明できる。
		2	線形探索、二分探索、ハッシュ法、二分探索木、文字列の探索など基本探索手法を理解している。
基本アルゴリズム	基本アルゴリズム	1	線形探索、二分探索、ハッシュ法、二分探索木について知っている。
		4	整列の基本アルゴリズムについて、その概念と特徴を十分に理解しており、卒業研究等でプログラムを作成する際に、目的に沿った適当な選択を行い、利用することができる。
		3	整列(ソート)などの基本アルゴリズムを、限られた条件下で使うことができる。処理すべき情報を適切なデータ構造として定義し、処理のための効率よいアルゴリズムを記述できる。
		3	整列、グラフアルゴリズムなどについて、動作を正しく理解し、具体的な入力例に対して机上でその動作を正しく追うことができる。それらのアルゴリズムの大部分を正しく記述することができる。
		2	バブルソート、文字列の照合、グラフ上の最小全域木の導出について実例を使って説明できる。
		2	代表的なアルゴリズムを他人に説明できる。代表的アルゴリズムは単純ソート、クイックソート、マージソート、ヒープソート、シェルソート、力任せ照合、NP法、B法などである。
		4	計算量とO記号によるその分析について、その概念を十分に理解しており、卒業研究等でプログラムを作成する際に、それを利用して効率の立場からプログラムの効率の良し悪しを比較することができる。
		3	最悪/平均計算量、O記法を理解する。漸化式を用いて計算量を評価できる。平均計算量の解析ができる。ソートの計算量の下限を理解する。
		3	設計したアルゴリズムの効率を基本的な解析技法の範囲で解析することができる。
		2	計算量の上限と下界、その評価法、および、計算量の漸近計算量(Oオーダー記法)について説明できる。
2	計算量とO記号によるその分析について、その概念を説明することができる。		
1	計算の時間について違いがあることを知っている。		

大項目	小項目	達成度区分	学生の達成度に関する具体的記述
コンピュータシステム構成とアーキテクチャ	数とデータの表現	4	2の補数系、1の補数系について、表現形式を理解しその計算が行え、浮動小数点については、その表現形式、誤差の計算が行え、誤差によって生じる問題とその対策法を説明できる。
		3	数値/非数値データの定義を説明できる。データの基数変換を行なうことができ、プログラミン等その知識を応用することができる。
		3	2進数、固定小数点、浮動小数点、文字コードについて基本的な説明ができ、その概念を用いてプログラムを作成できる。
		2	2の補数系、浮動小数点についてその計算機内での表現形式と計算のメカニズムが理解できる。浮動小数点については、誤差の意味が理解できる。
		2	数とデータについての種々のコンピュータ内部での表現方式を列挙できる。
		2	2進数、浮動小数点、文字コードについて理解しており、それぞれの具体例を説明できる。
		4	音声信号・画像・3次元物体のデジタル表現、データ圧縮・コンピュータグラフィックスのアルゴリズムの理解に基づいて、基本的な応用プログラムを実装できる。
		3	デジタル情報の基本的な符号化法について、目的と原理を知って使い分けができる。
		3	デジタル表現、標本化・量子化、データ圧縮などについて理解し、説明ができ、課題に対して具体的に実行できる。
		3	音声信号・画像・3次元物体のデジタル表現を理解し、与えられたデータ圧縮・コンピュータグラフィックスのアルゴリズムを実装するプログラムが書ける。
		2	画像をどのようにデジタル化して取り込み、表現するかについて、データ構造レベルで理解している。画像データに対してどのような変換処理を適用することにより、どのようなデータを得ることができるかについて理解している。
		2	アナログデータからデジタルデータへのサンプリング、データ圧縮、データの暗号化などの概念を理解している。
		4	簡単な計算機であれば回路設計ができる。パイプライン制御のメカニズムを理解しており、RISCマシンとCISCマシンのちがいとその特徴が説明でき、ある程度現実のプロセッサの性能をアーキテクチャの視点から評価できる。
		3	組み合わせ論理回路・順序論理回路の設計ができ、アドレッシングモード、割り込み、仮想記憶・キャッシュメモリなどのメモリ階層構造を説明でき、並列分散処理を実装するプログラムが書ける。
プロセッサ・アーキテクチャ	3	組み合わせ論理回路、順序論理回路の設計方法を理解している。機械語命令の実行過程を通してCPUの動きを信号レベルで理解し、与えられた問題に対して必要な機械語命令を正しく選択することができる。	
	3	組み合わせ論理回路、順序論理回路、命令セット、アドレッシング、割り込み、メモリ階層、キャッシュメモリ、並列処理について理解している。限られた条件の下で、組み合わせ論理回路および順序論理回路を設計できる。また、モデルコンピュータの下で、命令セット、アドレッシング、割り込み、メモリ階層、キャッシュメモリ、並列処理に対して、それぞれの仕組みを説明できる。	
	3	与えられた仕様の組み合わせ論理回路または順序論理回路を設計でき、その回路をハードウェア記述言語で記述できる。レジスタ、命令セット、アドレッシングの概念を解し、それらの仕様が与えられれば、簡単なアセンブリ言語または機械語プログラムを理解できる。プロセッサの構造と設計を理解し、与えられた命令を実行する際のデータフローを理解できる。	
	2	組み合わせ論理回路が設計でき、順序論理回路の動作が解析でき、アドレッシングモード、割り込み、仮想記憶・キャッシュメモリなどのメモリ階層構造を説明できる。	
	2	ノイマン型コンピュータがどのように実行されているか知っている。コンピュータアーキテクチャのトレードオフ論争の内容や、コンピュータアーキテクチャの変遷について知っている。並列実行過程（パイプライン処理、命令レベル並列化）やメモリアーキテクチャ（仮想メモリ、キャッシュ）の個々の技術について概説できる。	

大項目	小項目	達成度区分	学生の達成度に関する具体的情状
コンピュータシステムの構成とアーキテクチャ	オペレーティングシステム	4	プロセスの概念を理解している。マルチプログラミング環境にする目的とそとの問題点について理解し、構成方法を提案できる。各種のスケジューリングアルゴリズムについて具体例に対して実行評価できる。ファイルシステムの構成法について理解し、具体的なデータ構造を提案できる。逐次アクセス・ランダムアクセスのファイルについて構成方式や領域管理の方法について理解し、データ構造を提案できる。ディレクトリの必要性について説明でき、データ構造を構成できる。仮想記憶の構成とページング時のOSの役割について理解し、大筋の処理アルゴリズムを作れる。
		3	オペレーティングシステムの動作を理解した上で、オペレーティングシステムで発生する一般的な問題とその解決方法を理解する。さらに、オペレーティングシステムを利用する入門的プログラミングを行える。
		3	プロセス、マルチプログラミング、スケジューリング、ファイルシステム、ディレクトリ、仮想記憶、キャッシュメモリ、ページングについて理解しており、種々の方式について比較評価ができる。
		3	プロセス、マルチプログラミング、スケジューリング、ファイルシステム、仮想記憶、並行プロセス、クリティカルセクションなどのOSの重要な概念および使用方法を理解、説明、活用できる。
		2	典型的なオペレーティングシステムの概略について説明できる。
		2	unixの利用者としてシステムを活用できる。プロセスの並列実行、ジョブスケジューリング、仮想記憶、およびファイルシステムとアクセス法の構成について理解し、説明できる。
		2	マルチプロセスの管理、タスクスケジューリング、ファイルシステム、プロセス間通信、メモリ管理、仮想記憶管理の原理を説明できる。
		2	計算機システムにおけるOSの役割や基本概念（プロセス、スケジューリング、仮想記憶、ファイルシステム、保護など）について説明することができる。
		2	メモリ管理、プロセス管理、多重プロセスと同期処理、入出力制御、ファイル管理、仮想記憶、セキュリティと信頼性、性能評価などについて、基本的概念について理解している。
		4	交換技術、OS参照モデル、通信プロトコル等の基礎概念を説明することができる。LAN・WAN等を詳しく理解し実現に近い問題（卒業研究課題）に応用できるだけでなく、現実社会での問題解決に応用できる。
		4	ローカルエリアネットワーク（LAN）を詳しく理解し、実環境下におけるLANを構築することができる。
		3	OS参照モデルによるネットワークの階層モデルを理解しており、コンピュータネットワークに関する基礎的な知識を修得している。またコンピュータネットワークを利用した課題を実行できる。
		情報ネットワーク	コンピュータネットワーク
3	ネットワークの階層化方式、各階層の機能および階層間の関係を理解している。ローカルネットワークなどのネットワーク構成について理解し、指示のもとでアプリケーション・プログラミングの開発に反映させることができる。		
2	<ul style="list-style-type: none"> ・OS参照モデルを理解している ・TCP/IPモデルを理解している ・上記モデルとネットワーク機器との関連を理解している。 		
2	OS参照モデルの各層について説明できる。通信プロトコルの概念について理解しており、概要を説明できる。		
2	OS参照モデル、および、インターネットプロトコルスイートが説明できる。代表的なIWCプロトコルが説明できる。		

大項目	小項目	達成度区分	学生の達成度に関する具体的情状
情報ネットワーク	インターネット	4	TCP/IP プロトコル、電子メールや WWW 等のインターネットアプリケーションの原理について説明でき、サーチエンジンやネットワークワークモニタなどのツールを利用することができる。また、PCのネットワークに関する設定ができ、ネットワークプログラムの作り方を理解しており、卒業研究などに応用できる。
		3	インターネットの基本的な経路方式を理解し、電子メール、WWW、ファイル転送、サーチエンジンなどの活用や設定ができる。
		3	仕組みを理解した上で電子メールの送受信、WWW 閲覧、WWW 情報発信、サーチエンジンの使用ができる。
		3	IP アドレスや DNS による名前解決、HTTP (WEB) などのプロトコルを理解している。またソケット API を用いたプログラミングができる。
		2	IP アドレス、IP、TCP、UDP、DNS などが説明できる。ルーティング、輻辳制御の用語の意味がわかる。電子メール、WWW の仕組みが説明できる。
		2	TCP/IP 参照モデルの各層の機能と具体的なプロトコルを説明できる。IP アドレスの扱い方、クライアント・サーバモデルに基づく通信システム特性を説明できる。インターネットワーク上に構築されている応用プログラムの機構を説明できる。
		4	暗号、認証、ファイアウォールなど、ネットワークセキュリティの仕組みについて理解し、指示のもとで適切なアプリケーションを使用し設定することができる。
		3	ウィルス、ネット詐欺などの犯罪、ファイアウォールなどについてそのしくみを理解し、自分で防御策を施すことができる。また、公開鍵暗号や認証方式などについて理解している。
		3	暗号技術、認証技術について説明することができる。また、ファイアウォール、VPN について原理を説明することができる。また、自分の PC について、OS やアップリケーションなどのセキュリティに関する設定をすることができる。
		2	ファイアウォールの機能と、そこで用いられる技術について概要を説明できる。暗号と認証の必要性と有効範囲について概要が説明できる。
		2	暗号や認証等に関する基礎的知識を有している。RSA 暗号等の公開鍵暗号の概要を知っている。
		2	情報セキュリティの概念、暗号・認証・デジタル署名・生体認証・アクセス制御・コンピュータウイルスなどについて知識として理解している。
		3	指示に従って、ウォータフォールモデルで簡単なソフトウェア開発課題を遂行できる。
		ソフトウェア設計	ソフトウェアのプロセス
2	ソフトウェアのライフサイクルの概念を理解している。		
2	ソフトウェアのライフサイクルの各段階の内容を具体例を挙げて説明できる。		
1	ソフトウェアライフサイクル、ウォータフォールモデル、スバイラルモデルについて概要を知っている。		
1	ソフトウェアのライフサイクル、ウォータフォールモデルなど講義で聴いたことがある。		
4	ソフトウェアのライフサイクルについて理解している。オブジェクト指向プログラミングについて理解している。与えられた問題に対して、適切にクラス設計をして、プログラムを作成できる。		
3	指示に従って、オブジェクト指向プログラミング課題を遂行できる。		
3	モジュール設計、オブジェクト指向等の概念を用いて演習課題を解決できる。		
2	ソフトウェアの設計・作成の過程で用いられる基本的な考え方や技法について概略を説明できる。		
2	モジュール設計やコンポーネント、再利用などの概念やオブジェクト指向プログラミングについて理解している。		
データの設計と管理	データの設計と管理	1	設計技法にはどのようなものがあるのかを知っている。
		1	データ抽象化などについて講義で聴いたことがある。
		4	具体的にデータベースツール(と SQL)と言語(Java)を使って IP と URL にもとづき設計できる。
		3	関係データベース概要の理解と簡単な SQL プログラムが書ける。
		3	指示に従って、SQL を使用したリレーショナルデータベースの簡単な設計・管理の課題を遂行できる。
		2	データベースや記述言語に関して説明できる。
		2	データベース、Web 文書の基本を理解している。
		1	データベース設計のための実体関連モデル、関係モデル、SQL 言語を知っている。
		1	データベースに関する用語を知っている。

大項目	小項目	達成度区分	学生の達成度に関する具体的記述	
プログラミン グ言語の諸概 念	プログラミ ング言語の 基本概念	3	制御構造、データ型、関数、再帰、再帰、オブジェクト指向などを詳しく説明できる。さらに、それらを用いた簡単なプログラムを作成できる。	
		3	変数、データの型に関する基礎的な知識を習得している。また制御構造について理解し、ソートなどを含む基礎的なアルゴリズムを記述できる。さらに、オブジェ クト指向に関する基礎的な用語、及びオブジェクト指向の必要性について理解し、オブジェクト指向言語を用いて基礎的なプログラムを実装できる。	
		3	変数、データ型、制御構造、サブルーチン、パラメータ、再帰の概念が判り、演習課題等で使用できる。モジュール化、オブジェクト指向、並行プログラミング を知識として理解しており、人に説明できる。	
	言語処理系 の構成		2	・プログラミンング言語に用いられる再帰的な定義手法を理解している
			2	・プログラムをモジュール化する概念を理解している。
			1	0 言語に関する変数、データ型、制御構造、関数、構造体、ポインタなどの知識を学習し、その概要を説明することができる。
			3	変数、データ型、制御構造、サブルーチンなどの用語について知っている。
			3	コンパイラ、インタプリタ、オブティマイザ、BF、コンパイラ・コンパイラ、仮想マシン、ガーベッジコレクション等の概念を知識として理解しており、人に 説明できる。さらにBFによる文法記述と簡単なコンパイラ、インタプリタを作成できる。
			3	・再帰下降法に基づく構文解析プログラムの動作を説明できる。
			2	・ yacc/lex を使って、例を元にして、字句解析・構文解析系を作成することができる。
			2	言語処理系の基本的な構成を理解する。
			2	コンパイラ、インタプリタ、オブティマイザ、BF、コンパイラ・コンパイラ、仮想マシン、ガーベッジコレクション等の概念を知識として理解しており、人に 説明できる。
プログラミン グ能力	仕様の明確 化	2	コンパイラやインタプリタの役割と機能および実現方法を理解している。コンパイラの各フェーズのプログラミング技法を理解している。	
		4	仕様に記述すべきこととそうでないことを区別でき、仕様を明確化できる。	
		4	要求分析、仕様設計を応用できる。	
		3	外部仕様書、内部仕様書を作成できる。	
		3	与えられた設計仕様からモジュール仕様書（外部仕様、インターフェイス仕様）を作成できる。さらに、その内部仕様のうち処理概要（プログラムの振る舞 い）について、図式表現（フローチャート、UML図）ができる。	
		2	・ソフトウェア開発の要件分析等について理解している	
		2	・プログラム文書化技法を理解している。	
		2	データフロー図やユースケース図、クラス図などで要求分析を行うことを説明できる。また、モジュールの仕様として、引数の説明やファイルの入出力が必要で あることを説明できる。	
		2	要求を満たす仕様設計を行う方法を理解しており、具体例を挙げて説明ができる。	
		1	要求を満たす仕様設計について講義で聞いていて、その概要を説明できる。	
1	要求、仕様などの用語について知っている。			

大項目	小項目	達成度区分	学生の達成度に関する具体的記述
プログラミン グ能力	プログラムの 実装	4	与えられた仕様に基いて、プログラムを開発できる（コンパイラ等）。
		4	Cプログラミングについては、仕様に基きプログラムの実装を行うことができ、卒業研究に応用できる。
		3	小規模のプログラムであれば、与えられた仕様に適したプログラムを実装できる。平行程度のプログラムであれば、具体的な設計と説明が与えられれば、それに基 づくプログラムを実装することができる。
		3	構造化プログラミングやオブジェクト指向プログラミングの概念を用いて、与えられた演習問題を解くことができる。
		3	与えられた仕様に基いて、データ構造、制御構造、関数（サブルーチン）を用いてプログラムを記述することができる。
		2	<ul style="list-style-type: none"> ループなどの制御構造や配列などを用いたプログラムの動作を説明できる。 サブルーチンの動作を説明できる。 Java 言語でイベント駆動のプログラムの動作を説明できる。 Java 言語で継承を利用して簡単なプログラムを作成できる。
		2	構造化プログラミング、イベント駆動プログラミング、オブジェクト指向プログラミングの用語を知っておりその説明ができる。
		1	オブジェクト指向プログラミングについて知っている。
		1	構造化プログラミングやオブジェクト指向プログラミングの概要を述べることができる。
		4	仕様を満たすことを確認するためのテスト系列を作成し、実装の検証が行える。
		4	あらゆる可能性のテストデータを用意し、プログラムの検証、デバッグを実践的に実行することができる。
		4	仕様に通りプログラムの作成されたかどうかを調べ、誤りを修正できる。テストに必要な環境を調査でき、ほぼ自力でテスト仕様書を作成できる。
		3	プログラムのテスト、デバッグに関する基本的な知識は学習しており、演習プログラムに対して適用できる。
3	与えられた仕様から必要となるテストの内容を列挙できる。テストを実行し、誤りを発見して修正できる。		
3	Cプログラミングについては、演習課題のプログラムの検証・テスト・デバッグが実行できる。		
2	プログラムの正当性を検証する技法について理解している。		
2	<ul style="list-style-type: none"> ブラックボックステストとホワイトボックステストの違いを説明できる。 デバッグのコマンドについて、例を挙げて説明できる。 レビューの意義について説明できる。 ISO-9000について意義を説明できる。 		
2	作成したプログラムについて誤りが指摘されたとき、誤りの原因を調べ、修正するための方法を説明できる。		
2	仕様に通りプログラムの作成されたかどうかを調べるために必要なテスト技法（ブラックボックステスト、ホワイトボックステスト）について、説明できる。		
1	プログラムの正当性の証明やテストについて講義で聞いたことがある。		
1	プログラムの検証やテストにおいて何を行うのかを知っている。		
4	各種プログラミング環境を有機的・相補的に組み合わせ使用することができる。		
3	<ul style="list-style-type: none"> Windows・Linux 環境でエディタ・コンパイラを使い、プログラムを作成できる。 統合開発環境を使いプログラムを作成できる。 		
3	与えられた仕様からプログラムのソースコードの作成・編集、コンパイルおよび実行、デバッグを行うことが出来る。統合開発環境を利用できる。		
3	簡単なプログラムのソースコードの作成・編集、コンパイルおよび実行、デバッグを行うことが出来る。		
3	エディタを使ってプログラムを作成し、コンパイラで実行可能プログラムを作成し、実行できる。		
3	Eclipse エディタ、gcc、glibc、gdb、Eclipse 統合開発環境の基本的な使い方を理解し、Linux 環境でプログラムを作成し、必要に応じて、ディレクトリやファイルの整理 ができる。		
2	分割コンパイル、Make、Diff、RCS などの開発ツールについて説明できる。		
2	エディタやコンパイラを使ったプログラムの開発の概要について述べることができる。		
1	エディタ・コンパイラを起動できる。		
	プログラミン グ環境		

表 7. 2006 年度達成度調査結果

大項目	小項目	調査年度	平均レベルの学生の達成度					最低レベルの学生の達成度							
			5	4	3	2	1	0	5	4	3	2	1	0	
アルゴリズムとデータ構造	基本データ構造	2006	1	12	11	2	0	0	0	2	9	13	2	0	
		2005	0	16	6	3	0	1	0	0	12	12	1	1	
		合計	1	20	14	4	0	1	0	2	14	21	2	1	
	基本探索	2006	0	7	12	7	0	0	0	0	9	14	3	0	
		2005	0	5	15	5	0	1	0	0	9	14	2	1	
		合計	0	8	22	9	0	1	0	0	13	22	4	1	
	基本アルゴリズム	2006	0	8	15	3	0	0	0	2	7	15	2	0	
		2005	0	7	14	5	0	1	0	0	5	19	1	2	
		合計	0	10	24	6	0	1	0	2	10	25	2	2	
	アルゴリズムの効率	2006	0	4	16	6	0	0	0	1	6	16	3	0	
		2005	0	4	16	3	1	2	0	0	7	13	4	2	
		合計	0	7	23	8	0	2	0	1	10	22	5	2	
コンピュータシステムの構成とアーキテクチャ	数とデータの表現	2006	0	8	15	3	0	0	0	1	10	14	1	0	
		2005	0	6	16	3	1	0	0	0	4	19	3	0	
		合計	0	10	25	4	1	0	0	1	11	25	3	0	
	情報・メディアの表現	2006	0	8	9	5	0	2	0	1	6	15	1	1	
		2005	0	5	8	7	1	1	0	0	3	15	2	1	
		合計	0	10	12	10	1	2	0	1	7	21	4	2	
	プロセッサ・アーキテクチャ	2006	0	9	14	3	0	0	0	1	9	16	0	0	
		2005	0	8	12	6	0	0	0	0	6	18	2	0	
		合計	0	12	20	8	0	0	0	1	12	25	2	0	
	オペレーティングシステム	2006	0	0	8	1	0	0	0	0	2	7	0	0	
		2005	0	2	11	14	0	0	0	0	2	21	4	0	
		合計	0	1	12	11	0	0	0	0	2	18	4	0	
情報ネットワーク	コンピュータネットワーク	2006	0	7	9	8	0	0	0	1	8	15	1	0	
		2005	0	5	7	14	0	0	0	0	4	20	1	0	
		合計	0	7	14	17	0	0	0	1	10	26	3	0	
	インターネット	2006	0	7	10	7	0	0	0	2	10	12	1	0	
		2005	0	3	14	8	0	0	0	0	8	17	0	0	
		合計	0	8	16	13	0	0	0	2	11	25	1	0	
	ネットワークセキュリティ	2006	0	2	12	8	1	1	0	1	5	15	4	0	
		2005	0	2	6	14	1	2	0	0	0	19	4	1	
		合計	0	3	14	15	3	2	0	1	5	24	8	1	
	ソフトウェアの設計	ソフトウェアプロセス	2006	0	4	10	7	3	1	0	1	3	14	6	1
			2005	0	3	11	8	3	1	0	0	3	12	9	1
			合計	0	5	15	12	5	2	0	1	5	19	12	2
ソフトウェアの設計技法		2006	0	4	18	3	0	1	0	1	6	17	1	1	
		2005	0	5	15	4	1	2	0	0	5	14	6	1	
		合計	0	6	25	6	1	2	0	1	8	23	6	2	
データの設計と管理		2006	0	5	16	4	0	0	0	1	8	12	4	0	
		2005	0	3	16	5	1	2	0	0	9	7	8	2	
		合計	0	5	23	9	0	2	0	1	12	13	11	2	
プログラミング言語の諸概念		プログラミング言語の基本概念	2006	0	9	12	4	0	1	0	0	13	12	0	1
			2005	0	7	13	5	0	2	0	0	10	12	2	1
			合計	0	11	20	7	0	2	0	0	16	20	2	1
	言語処理系の構成	2006	0	3	13	7	0	0	1	0	4	15	3	0	
		2005	0	2	9	15	0	1	0	0	3	17	5	1	
		合計	0	5	16	16	0	1	1	0	6	24	6	1	

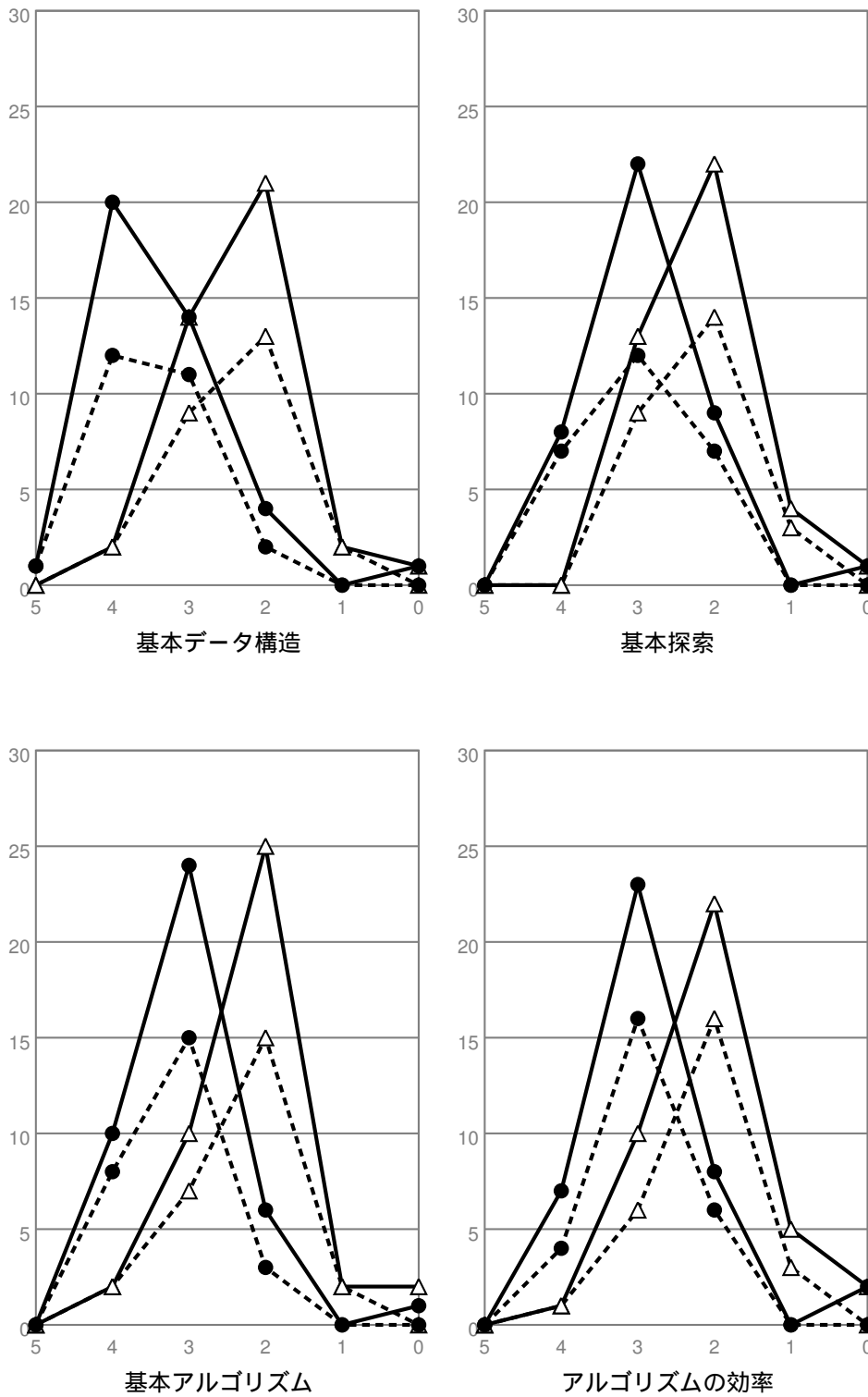
大項目	小項目	調査年度	平均レベルの学生の達成度					最低レベルの学生の達成度						
			5	4	3	2	1	0	5	4	3	2	1	0
プログラミング能力	仕様の明確化	2006	0	6	10	7	3	0	0	0	10	6	10	0
		2005	0	5	12	7	1	1	0	0	7	13	3	2
		合計	0	7	17	10	4	1	0	0	11	14	12	2
	プログラムの実装	2006	0	15	7	4	0	0	0	1	19	4	2	0
		2005	0	13	11	1	1	0	0	0	17	6	1	1
		合計	0	20	13	5	1	0	0	1	26	8	3	1
	プログラムの検証・テスト	2006	0	9	10	6	0	0	0	0	15	7	2	1
		2005	0	9	15	1	0	2	0	0	12	8	5	1
		合計	0	11	19	7	0	2	0	0	18	12	7	2
	プログラミング環境	2006	0	8	15	1	0	1	0	2	15	5	1	2
		2005	0	2	21	1	0	3	0	0	18	5	1	2
		合計	0	9	26	2	0	2	0	2	23	9	2	3
数学	離散数学	2006	0	5	14	6	0	0	0	0	9	15	1	0
		2005	0	3	20	4	0	0	0	0	11	13	2	0
		合計	0	6	25	8	0	0	0	0	15	21	3	0
	確率・統計	2006	0	2	20	2	0	0	0	0	9	13	2	0
		2005	0	2	20	3	0	1	0	1	8	13	1	1
		合計	0	3	30	4	0	1	0	0	13	20	3	1
Computer Science 専門領域	アルゴリズムとデータ構造	2006	0	7	10	1	0	0	0	0	7	9	2	0
	オペレーティングシステム	2006	0	10	15	9	1	0	0	0	13	21	1	0
	計算理論	2006	0	3	6	8	1	0	0	0	4	9	5	1
	コンピュータシステムの構成とアーキテクチャ	2006	0	5	11	3	0	0	0	0	8	10	1	0
	情報ネットワーク	2006	0	5	9	5	0	0	0	1	7	9	3	0
	数値計算・記号計算	2006	0	2	10	3	0	0	0	0	6	4	5	0
	ソフトウェア方法論・ソフトウェア工学	2006	0	4	8	6	1	0	0	0	6	9	4	0
	知的システム	2006	0	2	9	5	0	1	0	0	5	8	4	1
	データベース・情報検索	2006	0	5	8	4	0	0	0	2	4	8	3	0
	ヒューマンコンピュータインタラクション	2006	0	1	10	3	0	0	0	1	5	7	2	0
	プログラミング言語	2006	0	5	12	2	0	0	0	0	7	10	2	0
	原理の異なる複数のプログラミング言語	2006	0	8	9	2	0	0	0	0	5	12	2	0
Computer Engineering 専門領域	システムプログラムに関する基礎的項目	2006	0	1	5	6	0	0	0	0	2	6	2	2
	コンピュータシステム	2006	0	3	10	1	0	0	0	0	5	7	1	1
	情報通信	2006	0	2	5	5	0	0	0	0	1	7	3	1
	コンピュータ応用	2006	0	1	6	6	0	0	0	0	3	5	4	1
Software Engineering 専門領域	情報倫理・社会・法律・経済・安全	2006	0	2	4	2	0	0	0	0	4	5	0	0
	要求分析および設計	2006	0	0	5	2	0	0	0	0	2	4	2	0
	検証・正当性確認	2006	0	0	3	4	0	0	0	0	1	6	1	0
	実現・保守	2006	0	0	4	2	0	1	0	0	2	3	2	1
	プロジェクト管理等	2006	0	0	2	2	0	2	0	0	1	2	2	2
	コミュニケーション能力	2006	0	1	5	1	0	0	0	0	2	5	1	0
Information Systems 専門領域	データ管理	2006	0	1	3	1	0	1	0	0	2	2	1	1
	分析と設計	2006	0	1	4	1	0	0	0	0	2	3	1	0
	組織における情報システムの役割	2006	0	0	4	0	1	0	0	0	1	3	1	0
	情報システムを囲む環境	2006	0	0	3	1	1	0	0	0	0	3	1	1
	多様な情報システムの事例理解	2006	0	2	1	1	0	1	0	0	2	1	1	1
	問題形成・モデリング・プロジェクト管理	2006	0	2	1	2	0	1	0	1	1	2	0	2
	コミュニケーション能力・プレゼンテーション能力	2006	0	1	1	1	0	0	0	0	0	2	1	0
当該学部が特に力を入れている分野														

達成度区分 の平均	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46
2006 年度の 平均レベル	0	0	0	0	1	0	2	2	5	3	5	3	3	1	0	1	0	0	0
2005 年度の 平均レベル	0	0	1	1	1	1	0	1	8	7	1	5	2	0	0	0	0	0	0
2006 年度の 最低レベル	0	1	1	1	4	7	3	4	3	1	0	0	1	0	0	0	0	0	0
2005 年度の 最低レベル	1	2	0	1	2	9	7	4	1	0	0	0	0	0	0	0	0	0	0

回答数:

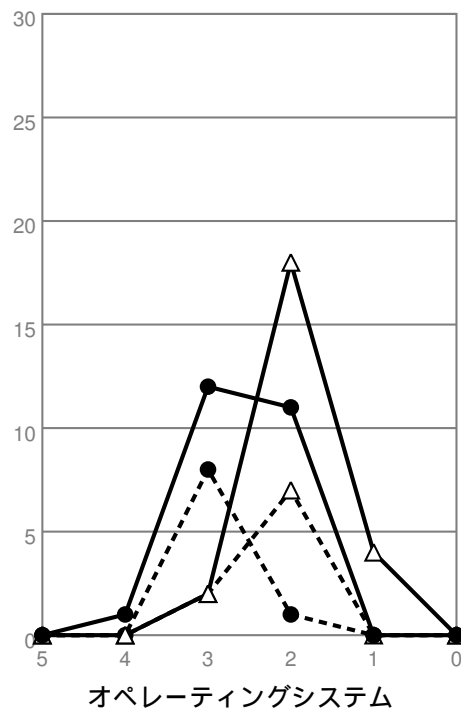
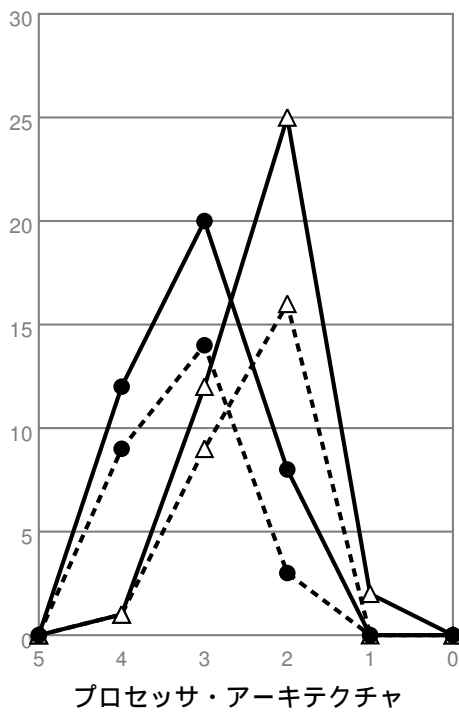
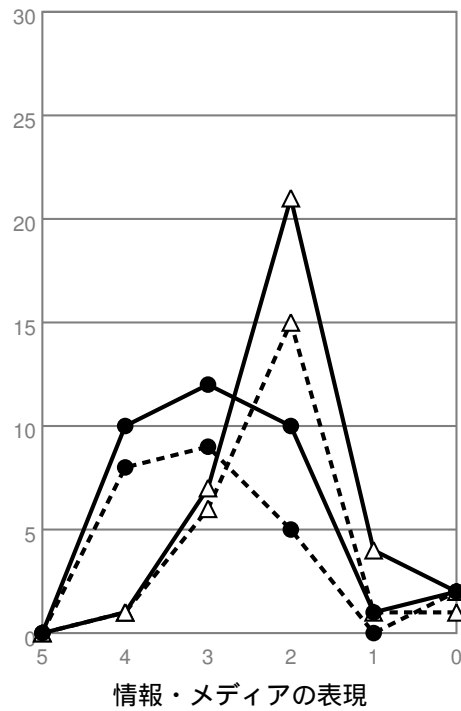
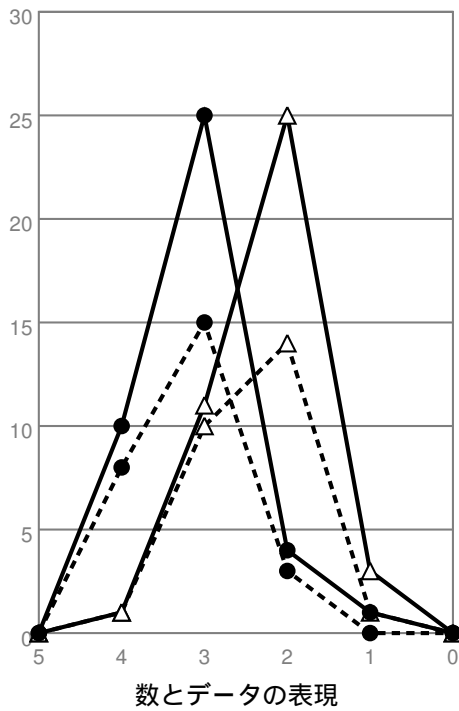
2006 年度	26 学科
2005 年度	28 学科
総数	41 学科

図2：2006年度および2005年度の達成度区分のグラフ
アルゴリズムとデータ構造



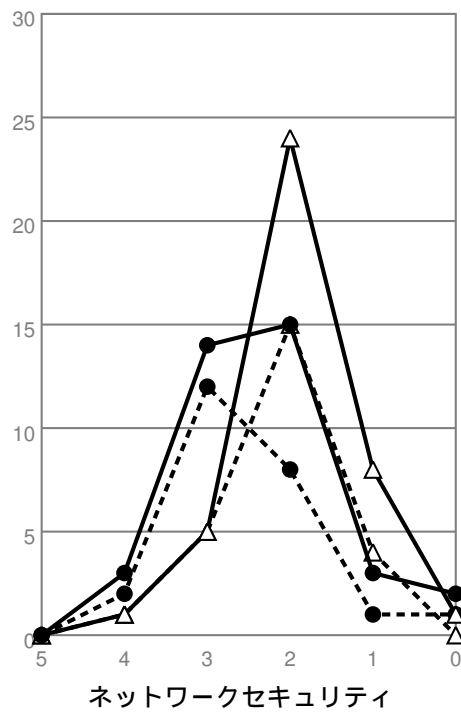
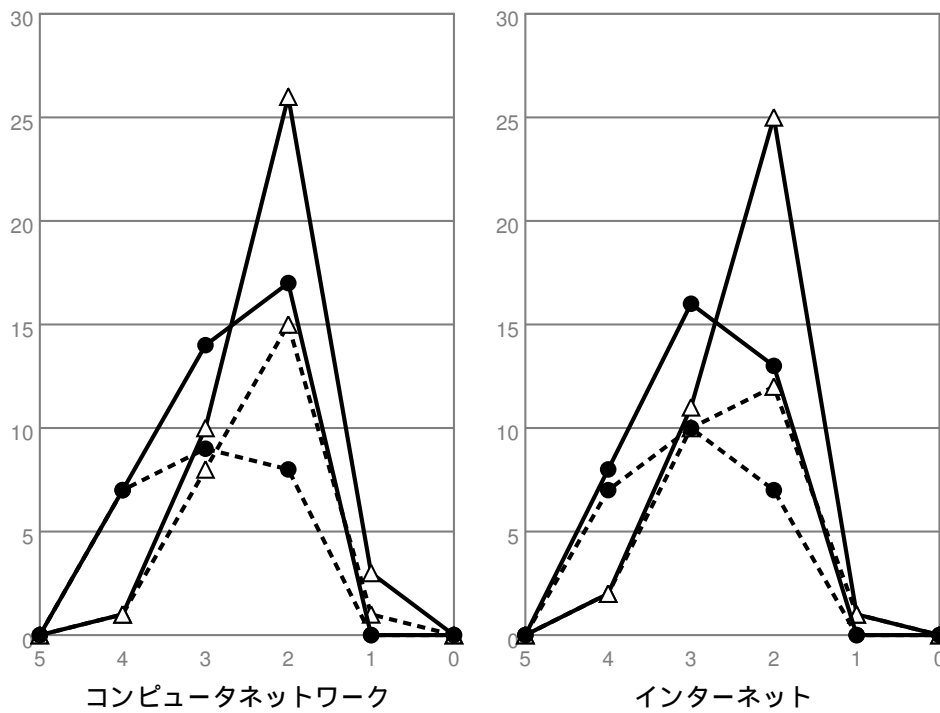
破線は2006年度、実線は2006年度と2005年度の合計、△は平均レベル、●は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

コンピュータシステムの構成とアーキテクチャ



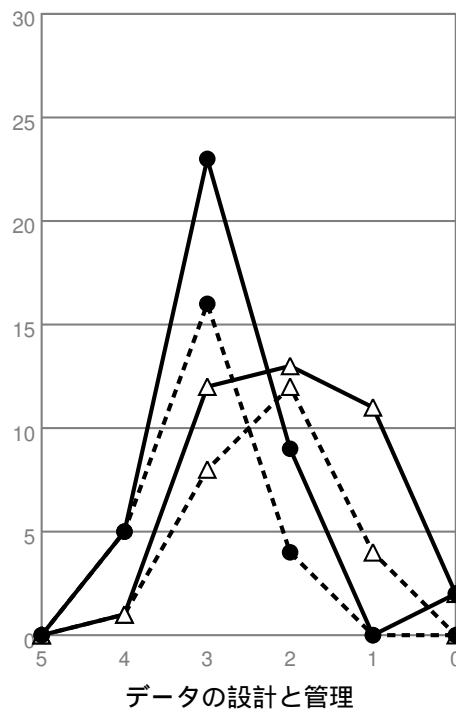
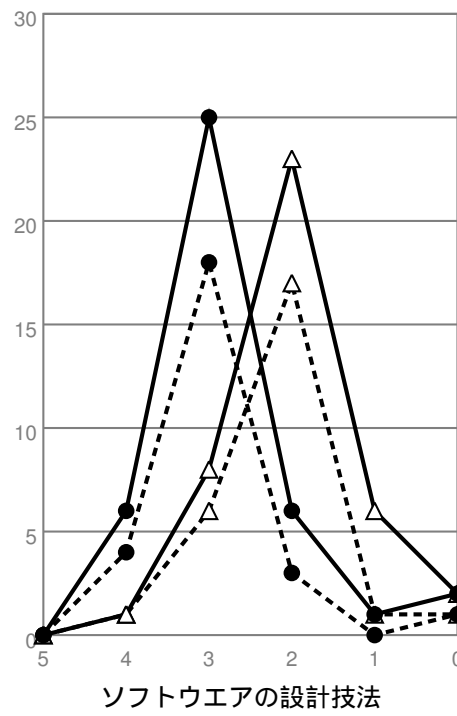
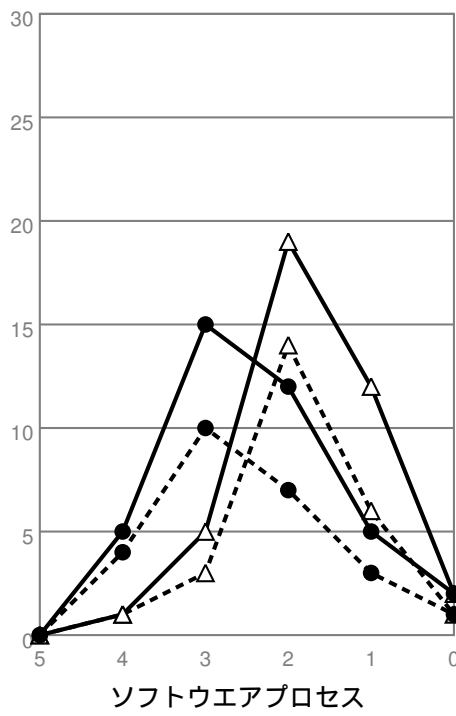
破線は 2006 年度、実線は 2006 年度と 2005 年度の合計、 Δ は平均レベル、 \bullet は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

情報ネットワーク



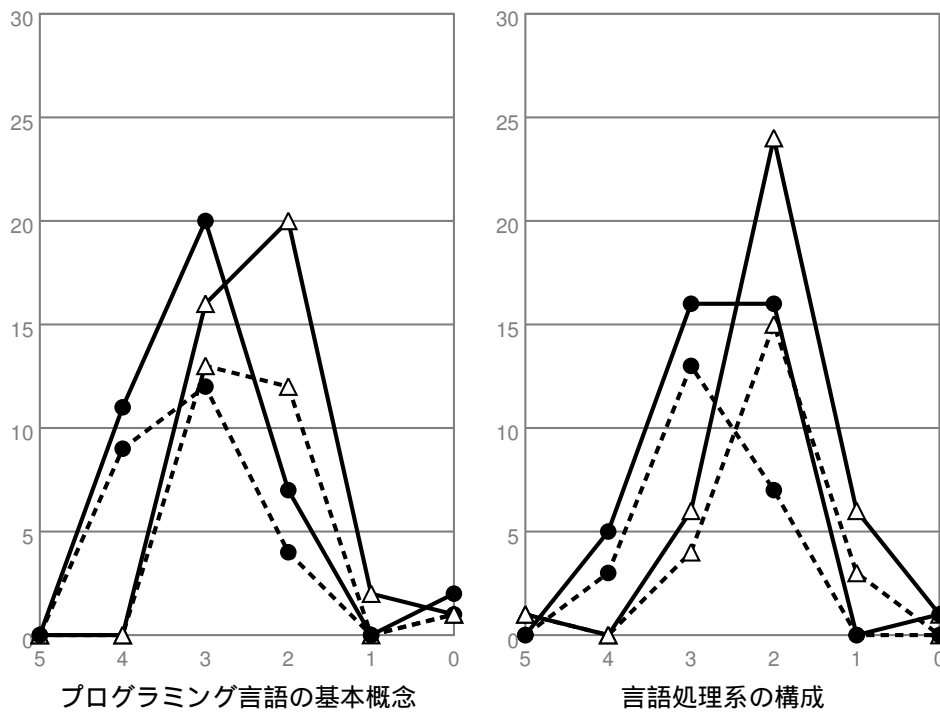
破線は 2006 年度、実線は 2006 年度と 2005 年度の合計、△ は平均レベル、● は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

ソフトウェアの設計



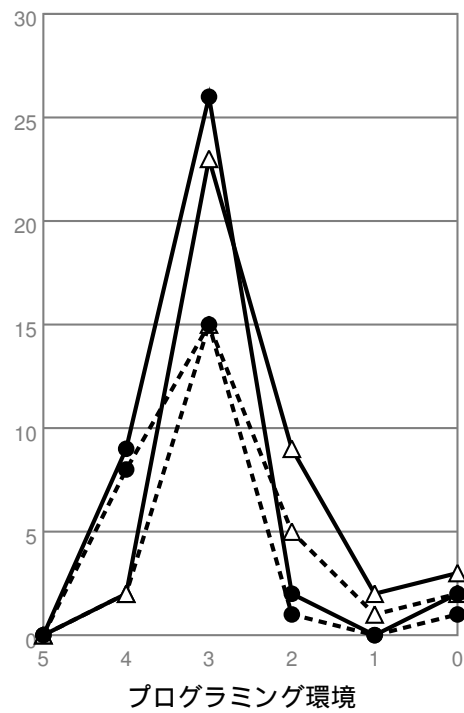
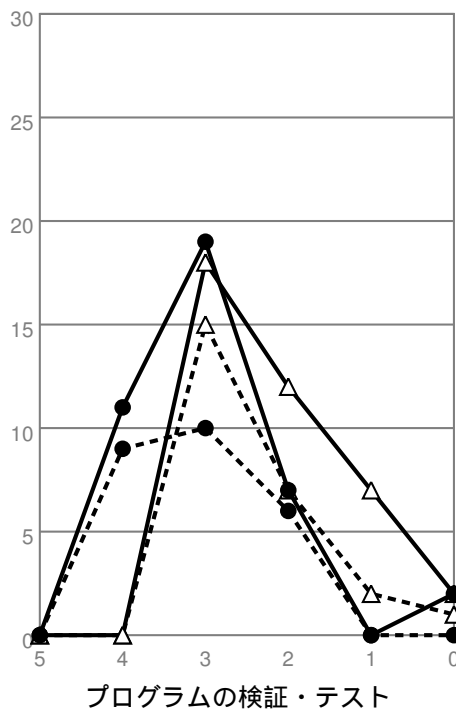
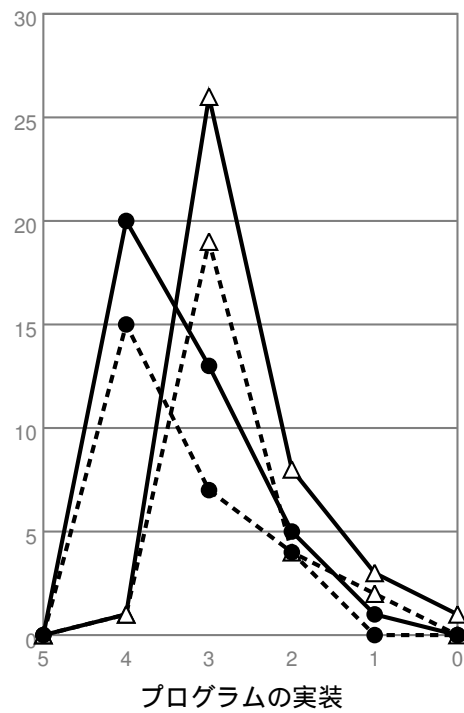
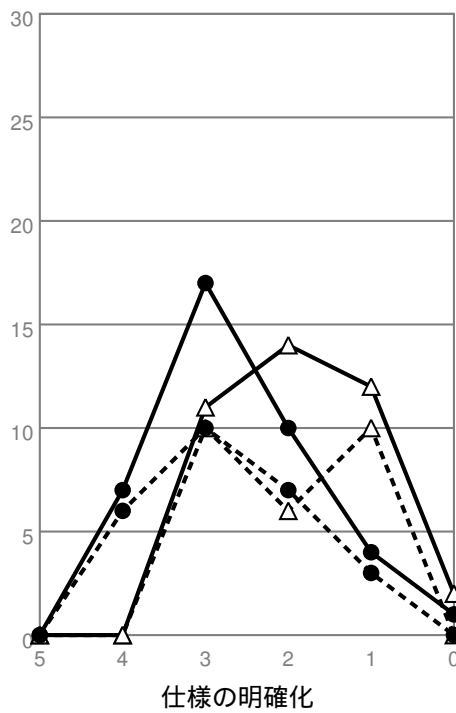
破線は 2006 年度、実線は 2006 年度と 2005 年度の合計、 Δ は平均レベル、 \bullet は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

プログラミング言語の諸概念



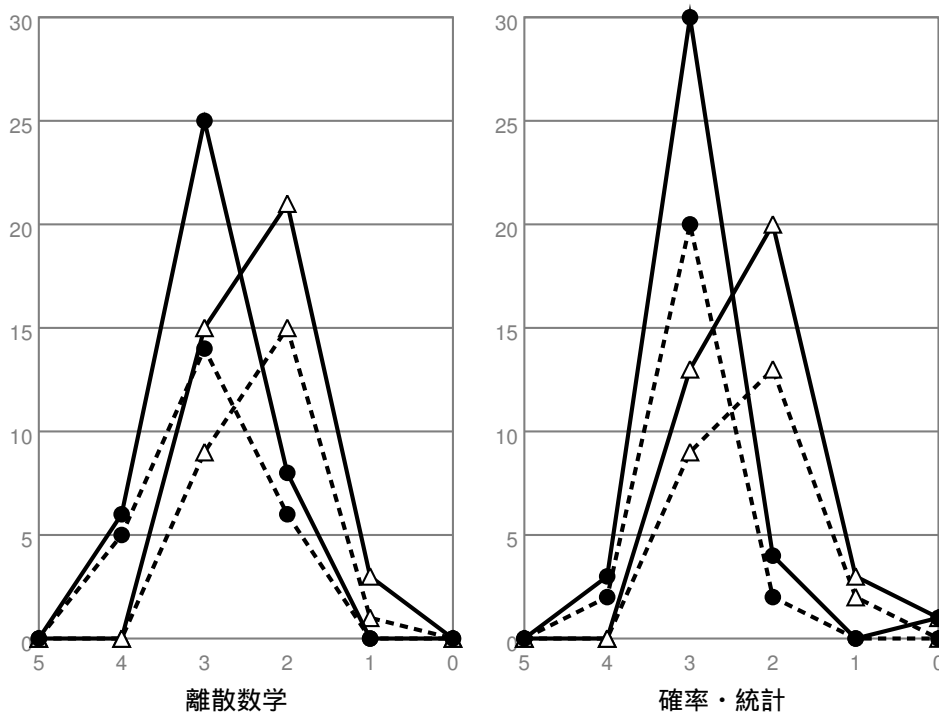
破線は 2006 年度、実線は 2006 年度と 2005 年度の合計、 Δ は平均レベル、 \bullet は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

プログラミング能力



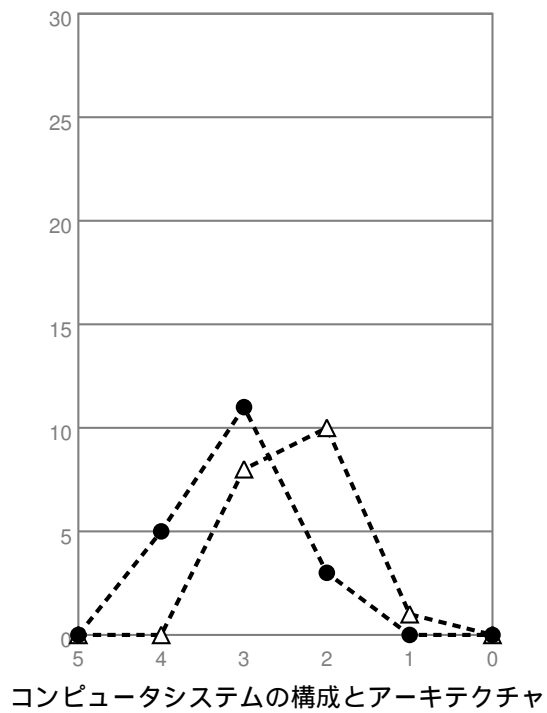
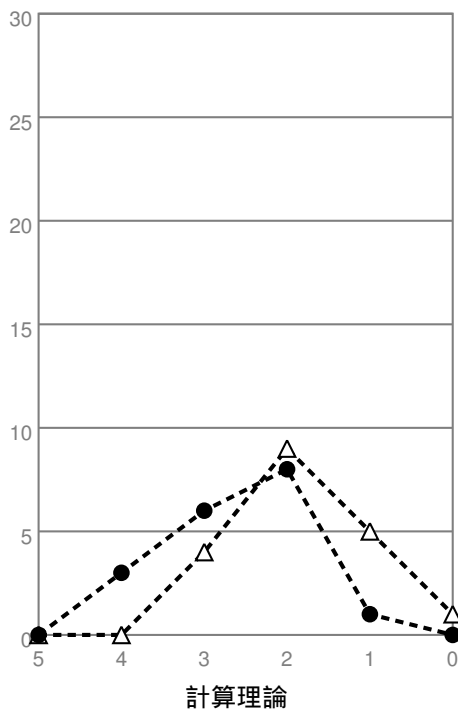
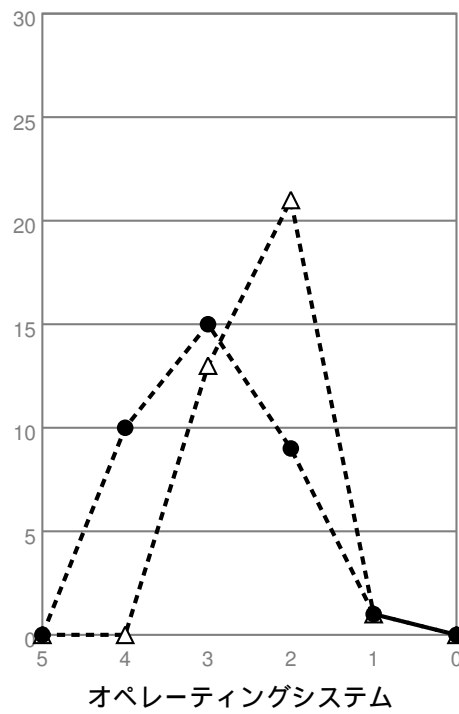
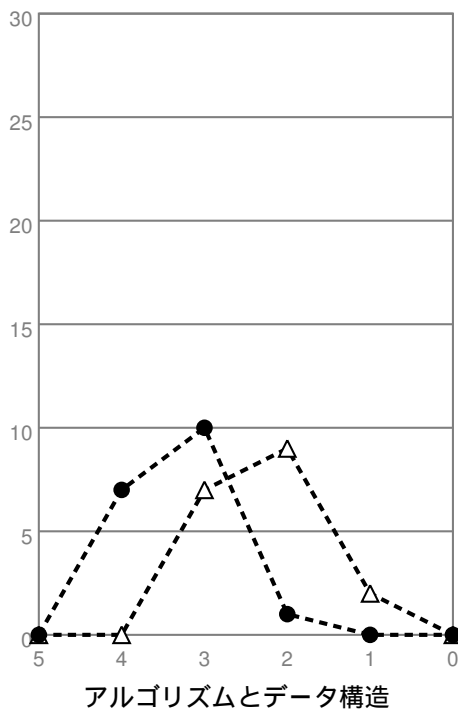
破線は 2006 年度、実線は 2006 年度と 2005 年度の合計、 \triangle は平均レベル、 \bullet は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

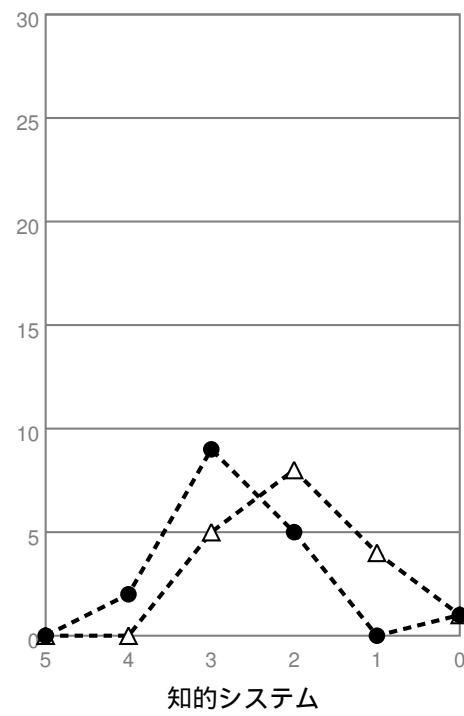
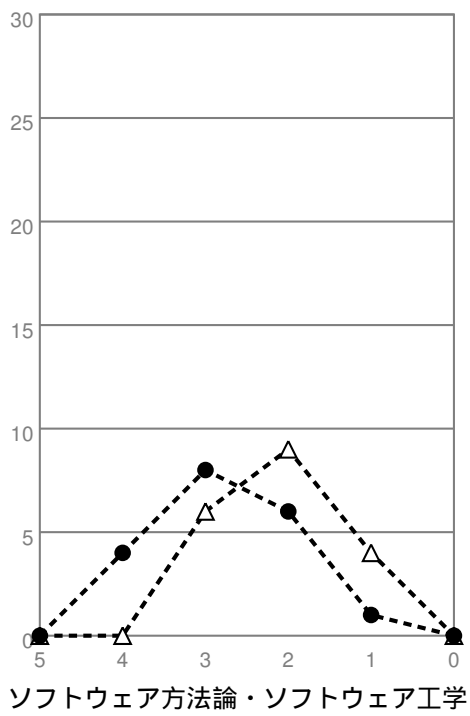
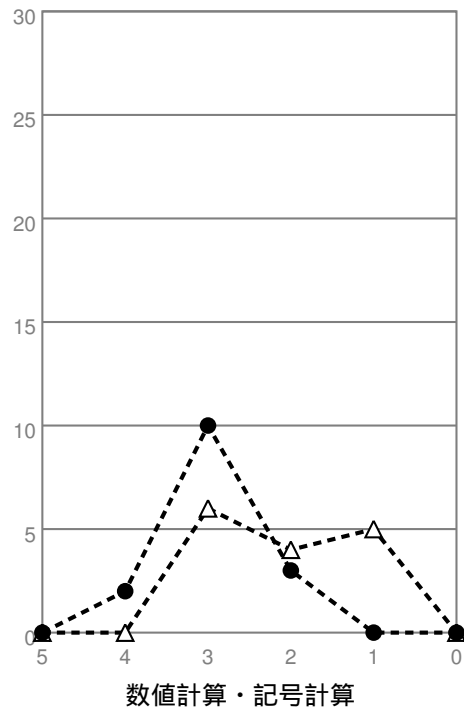
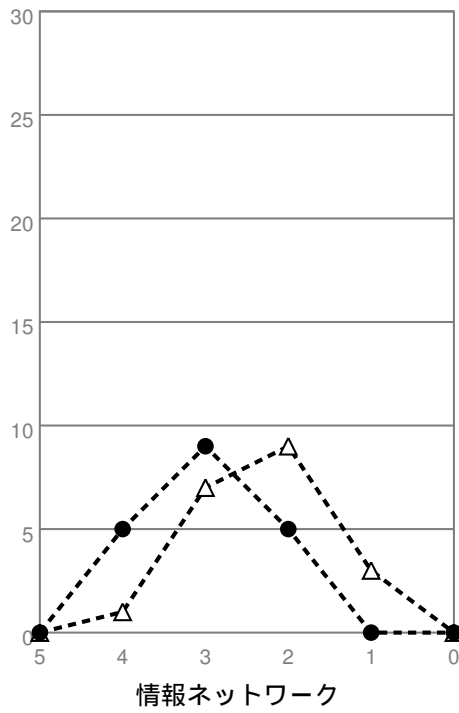
数学

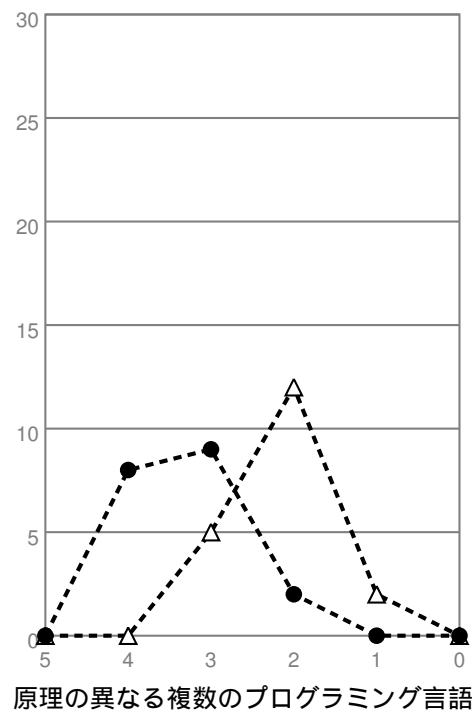
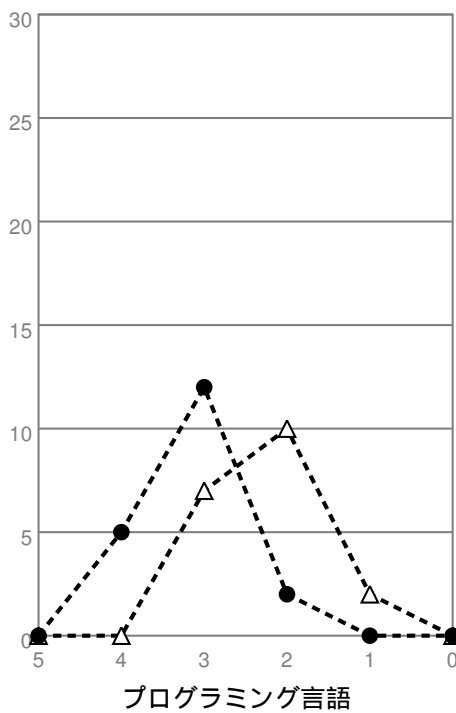
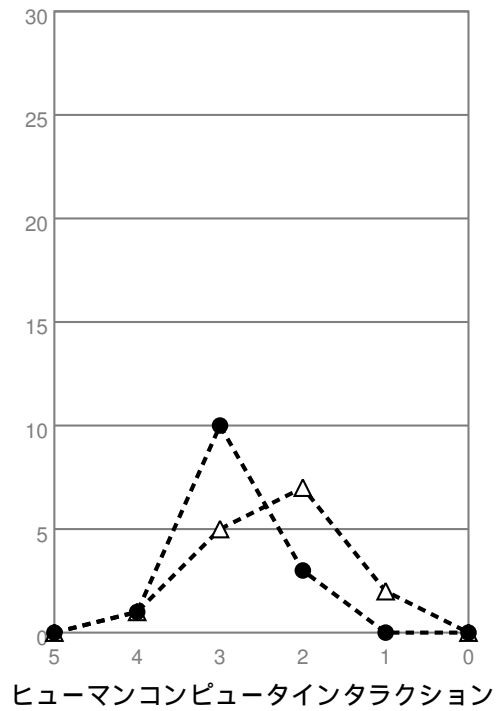
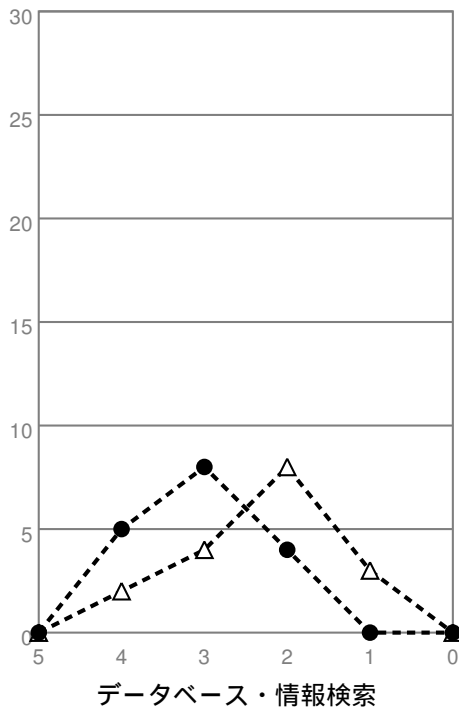


破線は2006年度、実線は2006年度と2005年度の合計、●は平均レベル、○は最低レベルを表す。
縦軸は学科数、横軸は達成度区分を表す。

Computer Science 専門領域

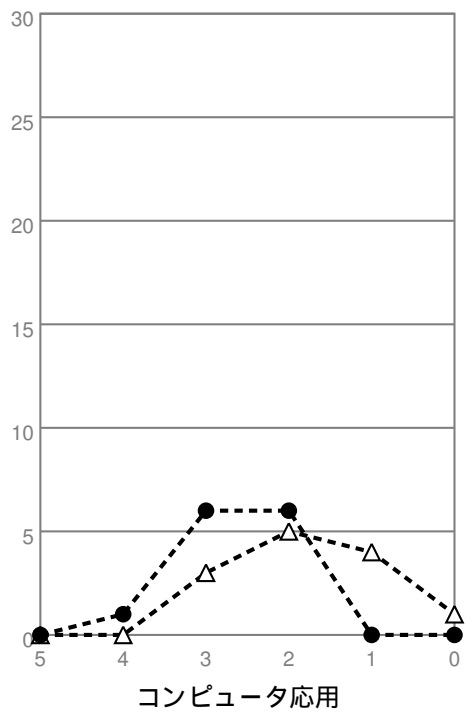
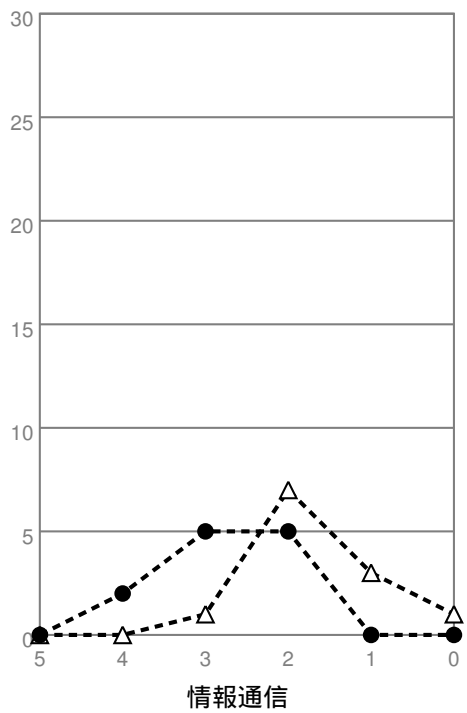
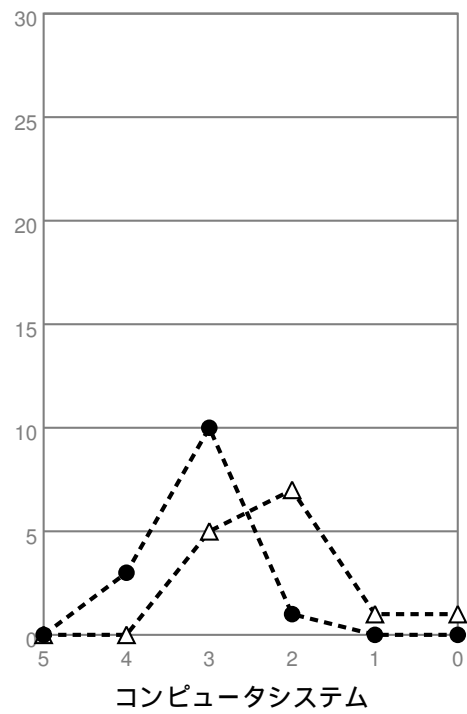
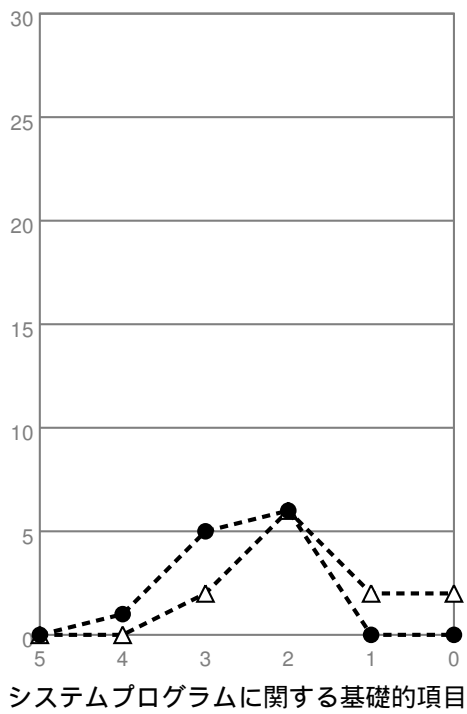






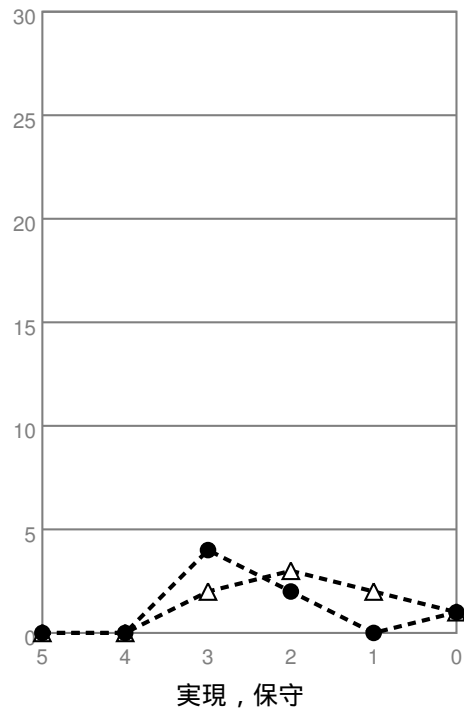
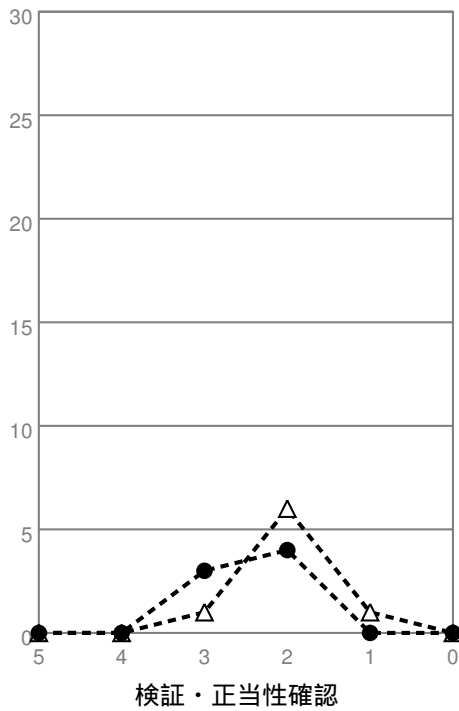
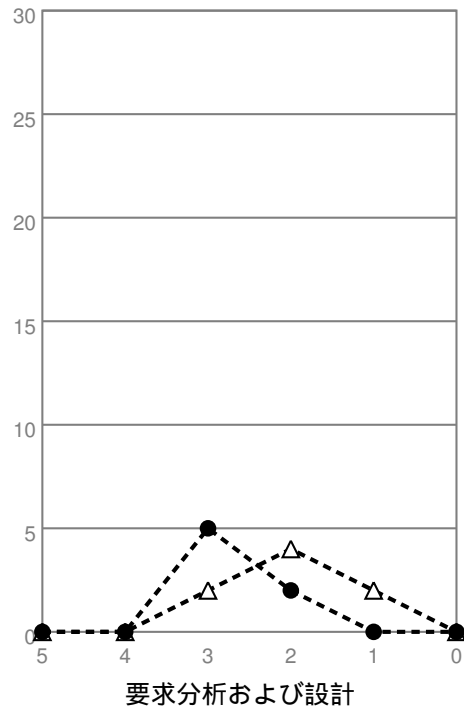
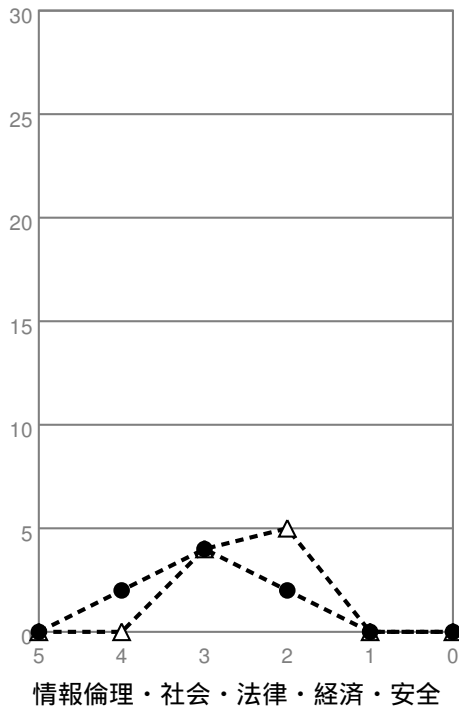
● は平均レベル, ▲ は最低レベルを表す。
 縦軸は学科数, 横軸は達成度区分を表す。

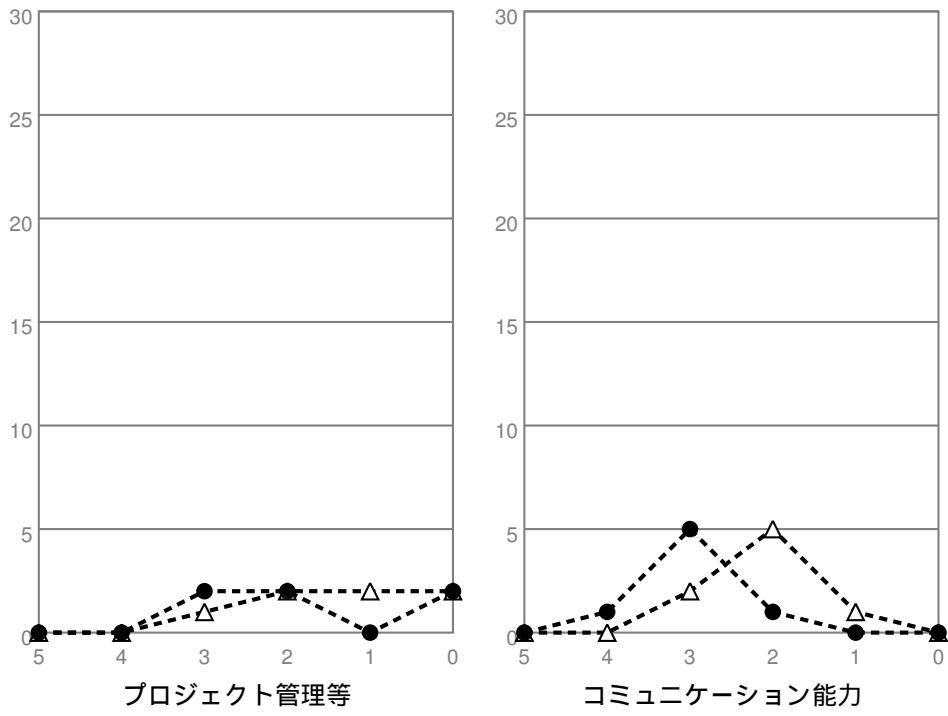
Computer Engineering 専門領域



● は平均レベル, ▲ は最低レベルを表す。
 縦軸は学科数, 横軸は達成度区分を表す。

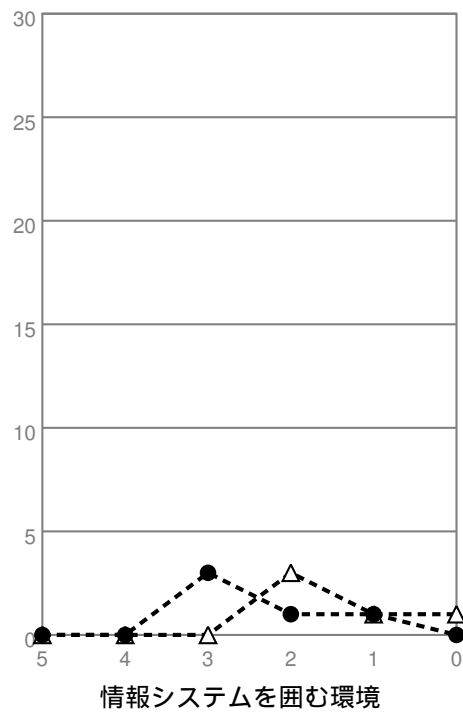
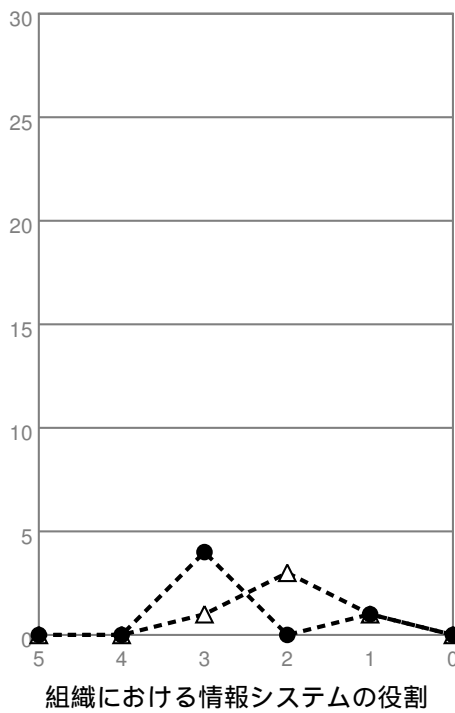
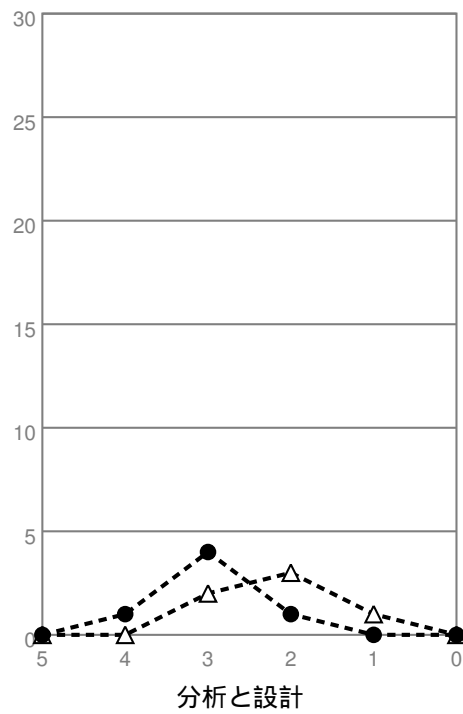
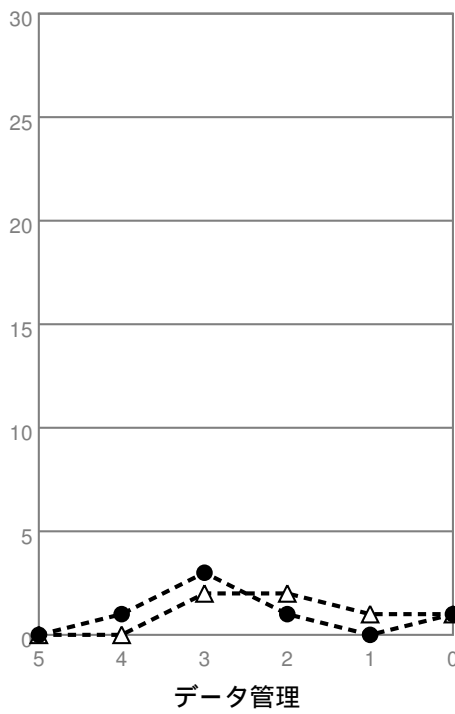
Software Engineering 専門領域

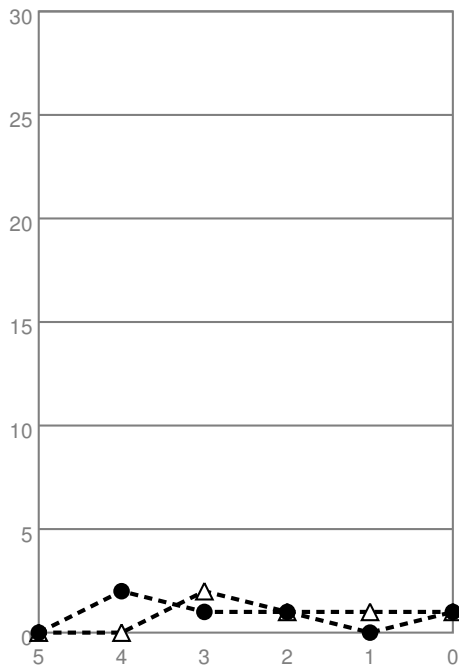




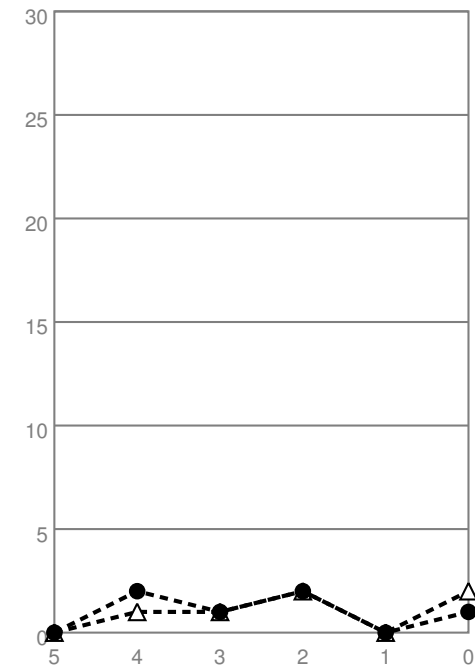
● は平均レベル, △ は最低レベルを表す。
 縦軸は学科数, 横軸は達成度区分を表す。

Information Systems 専門領域

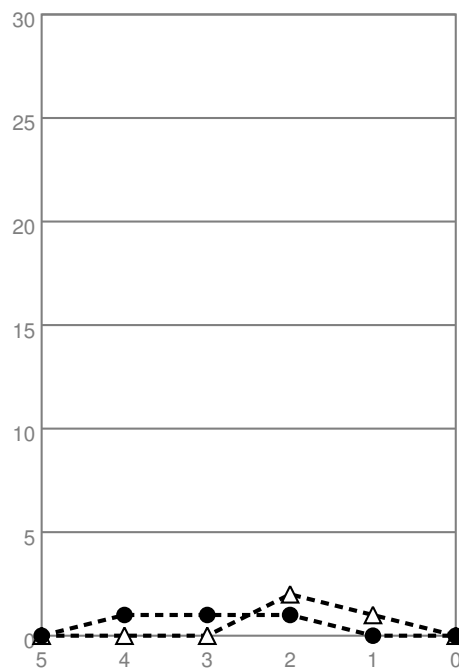




多様な情報システムの事例理解



問題形成・モデリング・プロジェクト管理



コミュニケーション能力・プレゼンテーション能力

● は平均レベル、△ は最低レベルを表す。
 縦軸は学科数、横軸は達成度区分を表す。