

알고리즘

- 2주차: 복잡도 -



데이터네트워크연구실
정진욱

jinwook1240@cs-cnu.org

알고리즘 실습 진행 방식

- <http://coding.cnu.ac.kr:8080>
- 매주 (보통) 10점: 출석 1, 과제 9
- 각 점수는 문제를 해결하고 제출, 심사받는 것으로 득점
- 출석: I/O 로만 이루어진 간단한 프로그램
- 과제: 알고리즘을 이용한 프로그래밍 문제 (코드 인터뷰 문제 난이도 조절)
 - 과제: 문제를 못 풀었을 경우 하루당 30%(기한 3일)
 - 보고서: 시간/공간 복잡도 해석 (STL 고려), 자신의 생각, 질문, 느낀점, 공유하고 싶은 문제
- 수업에서는 Python3 를 주로 사용 (Python3, Java, C, C++)
 - 질문은 어떠한 언어로 해도 됨. 하지만 Python 3 이외의 질문은 답변이 늦을 수 있음

실습 개요

- 실습 소개 및 알고리즘 설명: 20분
- 질의 응답을 포함한 문제 해결: 90분
 - 언어별 표준 라이브러리 웹 사이트만 허용
 - 일반 웹/검색은 X
 - STL문서에 익숙해지는 것이 좋음!
- 문제1(출석) (1): 최대 연속 부분합
- 문제2 (3): 배열 탐색
- 문제3 (3): 2D 최대 연속 부분합
- 문제4 (3): 회문 사전

Standard Library

- Python 3: <https://docs.python.org/3/library/>
- JAVA 8: <https://docs.oracle.com/javase/8/docs/api/> <https://dev>
- C: <https://en.cppreference.com/w/c>
- CPP: <https://en.cppreference.com/w/cpp>

| ID/ext↓ | external ID↑ | name↑ |
|---------|--------------|-------------|
| adb | ada | Ada |
| awk | awk | AWK |
| bash | bash | Bash shell |
| c | c | C |
| csharp | csharp | C# |
| cpp | cpp | C++ |
| f95 | f95 | Fortran |
| hs | haskell | Haskell |
| java | java | Java |
| js | javascript | JavaScript |
| kt | kotlin | Kotlin |
| lua | lua | Lua |
| pas | pascal | Pascal |
| pl | pl | Perl |
| sh | sh | POSIX shell |
| plg | prolog | Prolog |
| py2 | python2 | Python 2 |
| py3 | python3 | Python 3 |
| r | r | R |
| rb | ruby | Ruby |
| scala | scala | Scala |
| swift | swift | Swift |

실습 주의사항

- 실습시간
 - 정규 시간(1시간 50분)
- 실습 방법 : 실습==과제
 - 실습 문제 3~4개 출제
 - coding.cnu.ac.kr:8080 사이트에 주어진 시간 내에 코드 제출
 - e-learn.cnu.ac.kr 사이버캠퍼스에 보고서 제출
- 채점
 - 실습시간 내 통과 후 보고서 제출 마감시간 내에 제출해야 만점
 - 감점
 - 실습시간 내: 100%
 - 실습시간 후~당일 자정 : 70%
 - 이후 하루 30%씩 감점(실습당일 포함 3일간 제출 가능)
 - 유사도 높은 코드 검사 후 부정행위 적발 시 1회 경고, 2회 F
 - 부정행위 적발시 F
 - 보고서 : 과제기간+1일(목요일 23:59까지) 기한 내 감점 없음, 보고서 제출해야만 채점함

| 요일 | 시간 | 점수비율 (만점) |
|----|-------------|-------------|
| 월 | 14:00~15:50 | 감점없음 |
| | 15:50~23:59 | 70%(7점) |
| 화 | 00:00~23:59 | 40%(4점) |
| 수 | 00:00~23:59 | 10%(1점) |
| 목 | 00:00~23:59 | 보고서 제출기한 |

오늘의 주제: 복잡도

복잡도?

- 입력 n
- 프로그램이 얼마나 많은 연산을 수행하게 되는지?

```
1 statement;
```

```
1 for i in range(n):  
2     statement;
```

```
1 for i in range(n):  
2     for j in range(m):  
3         statement;
```

문제1(출석) [1]: 최대 연속 부분합

최대 연속 부분합

- Input
(숫자 리스트)
- Output
가장 큰 연속 부분합
- 이론강의에서 다룬 문제

예시 입력

1000 -1 1000

예시 출력

1999

예시 입력

1 38 3 -13 9 8 -99 24 2 4 10

예시 출력

46

문제2 [2]: 비교와 정렬

비교와 정렬

- Input
숫자 개수(2 이상)
숫자 배열
- Output
n # 최대 비교 횟수
정렬된 배열

| 예시 입력 |
|--------------|
| 4 8 4 2 3 |
| 예시 출력 |
| 6 2 3 4 8 |



| 예시 입력 |
|------------|
| 3 1 3 2 |
| 예시 출력 |
| 3 1 2 3 |

| 예시 입력 |
|--------------------|
| 5 10 2 58 -3 4 |
| 예시 출력 |
| 10 -3 2 4 10 58 |

문제3 [3]: 배열 탐색

배열 탐색

- Input

n m # 배열 길이 n , m 번째로 큰 수 구하기(0부터 센다)
(길이 n 의 배열) # 공백으로 구분

- 조건

n : 10000000 이하

m : 1000 이하

- Output

n 번째로 큰 수

- STL sort 사용 금지! 정렬 직접 구현할것

- 매우 큰 배열 : 시간 복잡도 고려할것

예시 입력

10 3
8 0 9 7 1 4 6 5 3 2

예시 출력

6

예시 입력

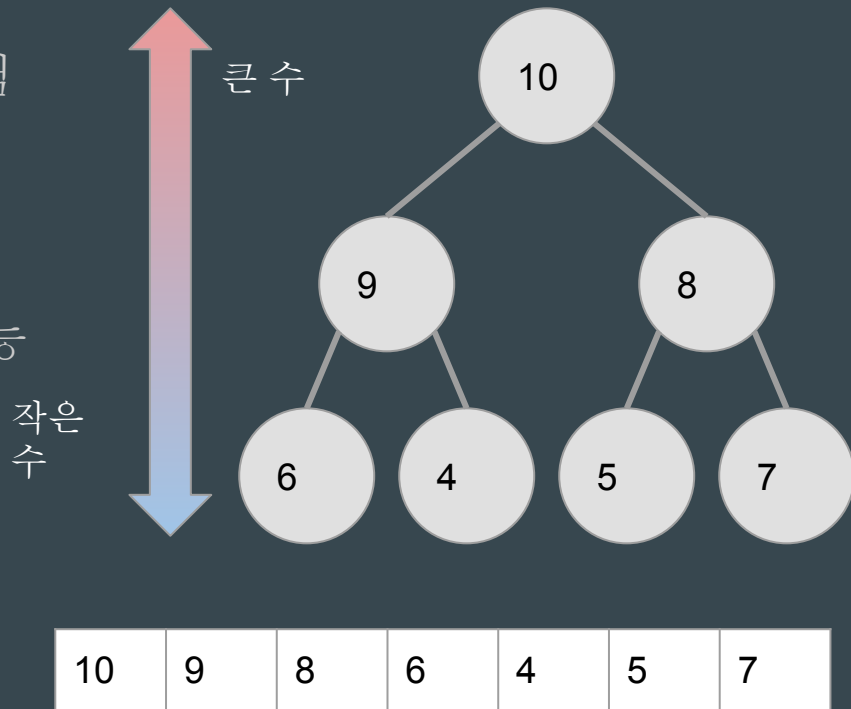
100 33
18 55 10 65 26 1 33 75 78 52
24 50 56 76 5 81 67 15 47 14
11 69 22 39 54 0 60 85 59 79
93 94 6 64 89 97 88 43 13 92
72 12 51 58 42 20 35 48 19 80
23 25 30 36 2 66 7 21 57 49 37
4 27 96 61 98 53 44 95 71 90
99 29 86 31 3 45 32 82 16 74
46 77 41 70 17 84 83 40 9 62
34 38 8 91 28 63 87 73 68

예시 출력

66

Heap

- 큰 수는 root로, 작은 수는 leaf로 보냄
- 파이썬 heapq 이용 가능
- 다른 언어도 heap 관련 STL 이용 가능



문제4 [4]: 2D 최대 연속 부분합

2D 최대 연속 부분합

- Input

n # 입력 크기 $n \times n$ ($n < 30$)

(n개의 공백으로 구분된 숫자) # 줄

- Output

가장 큰 연속 부분합

예시 입력

3
1 3 -2
3 -10 9
-6 9 4

예시 출력

13

예시 입력

10
3 -3 4 -1 1 -2 2 -5 -4 0
-1 -2 0 1 2 4 -4 -5 3 -3
-2 3 2 1 -1 4 -3 -5 0 -4
-1 -5 -2 0 3 1 4 -4 2 -3
1 0 -5 2 -1 3 -4 -3 4 -2
4 -3 -5 -1 2 0 -4 3 1 -2
-4 -1 -5 4 2 -2 3 1 -3 0
-3 1 -4 3 4 0 -1 2 -5 -2
-5 -1 -4 2 0 1 -3 3 -2 4
3 -3 4 -4 1 2 -1 -2 -5 0

예시 출력

34

참고
















제출 방법: 점수 확인

- <http://coding.cnu.ac.kr:8080> 에 접근하면 점수가 보임

Algorithm Week01 - Python Basic

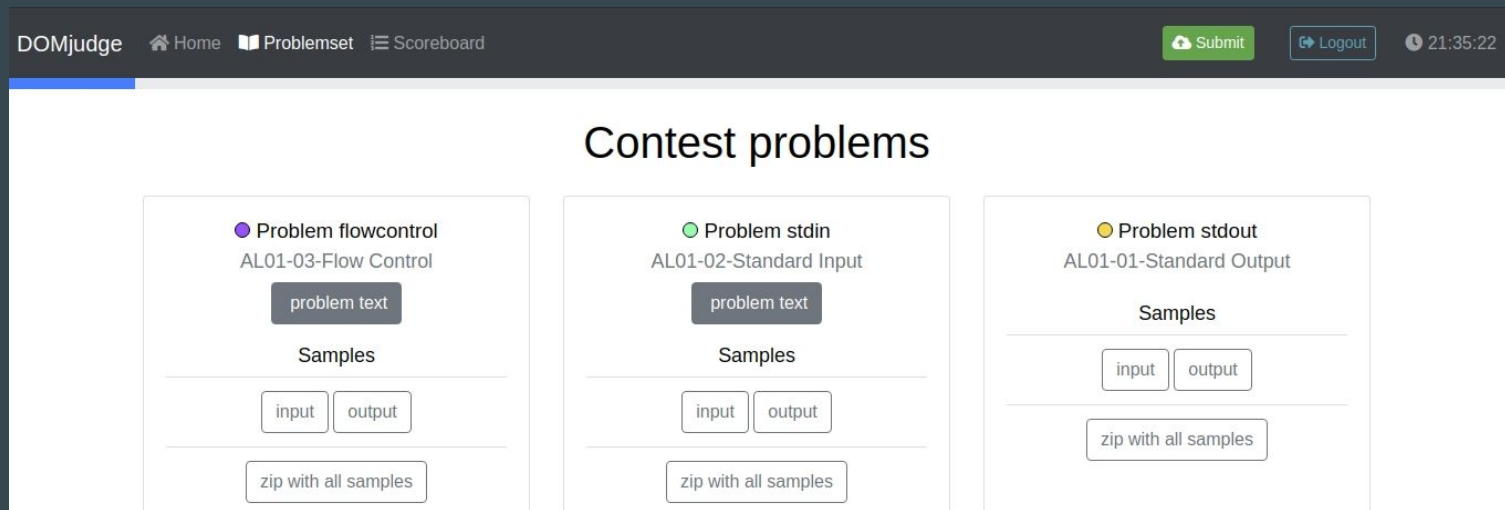
starts: 06:00 - ends: 06:00

Filter

| RANK | TEAM | SCORE | FLOWCONTROL  [5 POINTS] | STDIN  [4 POINTS] | STDOUT  [1 POINT] |
|---------|---------|-------|--|--|--|
| 1 | ♡ DNLab | 0 0 | | | |
| SUMMARY | | |  0  0  0  n/a |  0  0  0  n/a |  0  0  0  n/a |

제출 방법: 로그인 후 제출

- 로그인 하면 문제 셋 표시됨
- Submit 을 눌러 코드 제출 (*.py)
- 점수 표에서 득점 확인 가능



The screenshot shows the DOMjudge website interface. At the top, there is a navigation bar with links for Home, Problemset, and Scoreboard. On the right side of the navigation bar, there are buttons for Submit and Logout, and a clock showing the time 21:35:22. The main content area is titled "Contest problems" and displays three problem cards. Each card has a colored circle icon, a problem name, a "problem text" button, a "Samples" section with "input" and "output" buttons, and a "zip with all samples" button.

DOMjudge Home Problemset Scoreboard Submit Logout 21:35:22

Contest problems

Problem flowcontrol
AL01-03-Flow Control

problem text

Samples

input output

zip with all samples

Problem stdin
AL01-02-Standard Input

problem text

Samples

input output

zip with all samples

Problem stdout
AL01-01-Standard Output

Samples

input output

zip with all samples

질문

- 코드 피드백
 - <https://forms.gle/4B51eDpwovTUCXTM8>
 - 수업시간동안 코드 관련 질문은 이 링크로만 받습니다.
- 질문 메일(수업시간 외)
 - TA 이메일(jinwook1240@cs-cnu.org)
 - 제목 양식 : [AL20]제목
 - 본문이나 제목에 학번/이름 명시해주세요
- FAQ 문서
 - <https://docs.google.com/document/d/1ntR6GS1SI7dRbYlw-pu8uT8U65Wc-RtMj10IEpiapfU/edit?usp=sharing>
 - 질문 전 원하는 답변이 있는지 확인해보기

잊지 않아야 할 것) 소스코드 및 보고서

- 사이버 캠퍼스에 목요일까지 제출 (하루 뒤)
- 시간/공간 복잡도 해석 (STL 고려), 자신의 생각, 질문, 느낀점, 공유하고 싶은 문제
- 점수 확인 화면(이전 슬라이드 참고) 반드시 추가
- 문제를 해결하지 못 하였을 경우에도 자신의 소스코드 및 보고서 내용 제출
 - 코드 완성 정도에 따라 부분점수를 받을 수 있음!
- 채점 결과표:
 - 구글 스프레드시트로 공개 예정

만족도 조사

- <https://forms.gle/buE8jmoMv7sSVUmN7>
- 수업 품질 향상을 위하여 꼭 참여해주세요!