

# 2024 Spring Term Project, Data Structure

Dept. of AI Engineering, Sookmyung W. University

Issue Date: 2024.05.04(토)

Professor: Young-Ho Park

E-mail: [yhpark@sm.ac.kr](mailto:yhpark@sm.ac.kr)

- 제출 기한 및 사용 언어
  - **2024년 06월 02일 (일요일) 22시 00분 정각, snow board에 정시 제출 마감.**  
**No delayed submission, No e-mail submission (정상 제출자와 형평성 사유)**
  - 과제 프로그램 작성 관련 질문(문법, 실행 오류, 프로그램 방법 등)은 프로그래밍 과제이므로 공히 받지 않음.
    - ◆ 만약, 프로그램에 대한 가정이 필요하다면, 문서에 가정 사항을 명기하고 그대로 프로그램 하면, 해당 사안을 인정할 예정임.
  - 사용 언어: C 언어
- 배점
  - 20점 만점
  - 프로그램 copy의 경우는 **과목 최종 성적 'F'** (유사도 검증 프로그램에서 유사로 판단되는 제출물은 'F' 처리 예정이며, 프로그래밍 과제이므로 이 부정행위에 대해서는 이의를 제기할 수 없음, 인터넷 copy와 chatGPT 산출 프로그램 check됨, 이를 검증에 포함 예정)
  - **[프로그램 수행이 안된 경우]** 프로그램을 시도했으나, 결과적으로 제출 수준으로 작성을 하지 못한 경우, 합리적인 사유를 작성하고, 그 작성 내용이 합당한 경우, 그리고 아래 요청한 **(A)~(C)** 문서가 모두 있는 경우에 한하여, 각 문제 별로 최대 1점까지의 노력 점수 부여 가능함.
  - 아래 제출 문서 중 1개 누락되면 채점 및 증빙이 불가능하기 때문에 채점없이 0점 처리 예정임. 즉, **[주의]** 1점이라도 받으려 한다면, 일단 모든 문서는 존재해야 함.
- 최종 제출물 설명

제출할 최종 압축 파일 제목: **[2024 DS Proj -x분반] 학과(주전공)-x학년-학번-이름**

제출 파일 예시: **"[2024 DS Proj-2분반] 인공지능공학부-2학년-2311098-홍길동"**

- 총 4개의 과제임  
모든 과제를 다음의 방법으로, “4개의 과제를 1개의 압축 파일로 압축”하여, 스노우보드 제출란에, 제한된 시간 이전에 업로드 함.
- 압축된 파일 내에는, 4개의 각 과제에 대해, 서로 다른 폴더로 만들어진 4개의 폴더가 들어 있어야 한다. 각 폴더에는 문제 번호를 붙여서, 폴더 이름을 구분 가능하게 만들기 바란다. 해당 과제를 과제 번호가 붙은 해당 폴더 내에서 수행해야 한다. 채점은 폴더 별로 채점 점수를 부여한다. 4개 과제에 대한 폴더 이름의 예시는 다음과 같다:  
“과제1번”, “과제2번”, “과제3번”, “과제4번”
- [주의] 압축을 풀었을 때 위 예시와 같은 해당 폴더(“과제1번”~“과제4번”)가 없는 경우 채점하지 아니하고, 해당 문제는 0점 처리 한다.
- 최종 제출할 과제 폴더 별로, 다음 (A)~(C)의 소항목 파일, 3개씩이 각기 담겨 있어야 한다. 프로그램이 정확히 수행되어야 하고, 수행된 프로그램에 대한 다음 3개의 파일이 반드시 있어야 해당 문제에 대한 해당 점수를 부여할 수 있다.
- [주의] 즉, 3개 중 1개의 파일이라도 부족하면, 채점을 하지 않을 예정임.
- [주의] hwp, doc 파일만 허용, ppt안됨. (문서의 작성일 등을 파악하기 위함임.)
- 제출할 3개의 파일에 대한 설명
  - (A) 파일 제목: “(A) Program Code”  
.c, .h 파일들, 각 파일(프로그램)에는 최 상단에 주석 기능을 사용하여, 본인이 프로그램 저자임을 밝히는 (1) 학과, (2) 학번, (3) 이름, (4) 프로그램 파일명, (5) 프로그램 실제 작성일, 이렇게 5개항을 명기함.
  - (B) 파일 제목: “(B) 2\_descriptions”  
(각 함수에 대한 알고리즘 설명과 그렇게 구현한 이유 등을 제3자가 이해할 수 있도록 자세히 기술한다):  
**각 프로그램 기능에 대한 보고서**이다. 본 파일에는 다음 2가지 내용이 담겨 있어야 한다:

(1) 사용한 **각 함수**에 대한 설명 (Input, Output, 알고리즘 설명): 만약, 사용한 함수가 2개의 경우, 예를 들어, main()과 add()인 경우, 2개에 대해 각각 글로 설명하는 부분이다. 상기 (A)번 문서에 첨부한 프로그램에 대한 정확한 해석을 구체적으로 제시해야 한다.

(2) **[평가에 매우 중요]** 실행 성공 결과 화면 단계별 Capture (결과가 많은 경우 작은 화면으로 Capture해서 첨부할 것) - **없는 경우, 과제 수행 결과가 없는 것으로 판단함. 이것을 가장 중요시 볼 예정이니, step별 결과를 모두 첨부해야 함.**

■ (C) 파일 제목: “(C) Progress Report”:

(1) 투자한 순수 집중 시간 및 과제 결과로 본, 본인이 원하는 **문제별 요청 점수와 객관적 근거 제시:**

예시) 문제 1의 경우, 2점씩 4점으로 구성되어 있다. 이 경우 다음과 같이 보고서를 작성할 수 있다 예시는 다음과 같다:

**문제1 (A):**

주장하는 점수: **1점 (부분 점수 요구함),**

주장 근거: 프로그램은 자체에는 문제가 없어 보이나, 일부 결고에 대한 답이 틀리게 나옴. 이에 대한 스크린 샷을 첨부하였음. 기능에 대한 기술적인 설명은 충분하게 하였음.

**문제1 (B):**

주장하는 점수: **2점 만점 요구함,**

주장 근거: 프로그램 수행에 문제가 전혀 없고, 답도 정확히 나왔으며, 문제에서 요구하는 모든 내용을 구현하였고, 해당하는 함수에 대한 기술적인 내용을 전부 기술하였고, 해당 수행 과정에 대한 스크린 샷 증빙도 정확히 제출하였습니다. 뿐만 아니라, 많은 시간을 할애하여, 프로그램에 대한 내용 검증까지 마쳤으므로 만점을 요청할 수 있음

(2) (a) 도움을 준 학생, (b) 참고한 문헌, (c) 인터넷주소:

그 **참고 내용과 범위,**

**만약, 없으면 없다고 명기하고,**

**강의록을 참고한 경우도 강의록이라고 명기함.**

ChatGPT를 사용한 경우는 스스로 과제를 하지 않았기 때문에 0 점 처리 예정임.

단, chatGPT에서 어떤 단순한 도움을 받은 경우는 이를 기록할 것.

(3) Daily Report:

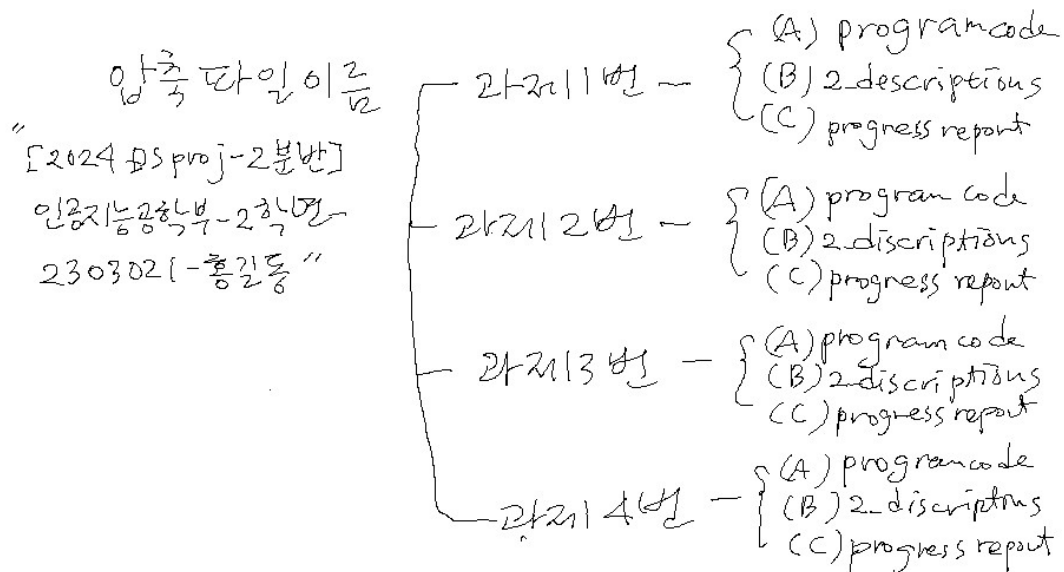
과제 수행 과정을 상세히 기록한, 과제 받은 후, "실제 착수일 ~ 제출일"까지 날짜 별 daily report (무형식, 즉 일기 형식 추천)

→ [주의] 상기 (1)~(3) 번호를 가능한 객관적으로 이해가 되도록 기술해야 함. 3개의 파일 중 1개라도 없는 경우는 본인이 과제를 실시하지 않았다고 판단하고, 해당 과제는 0점을 부여 할 예정임.

초기 0점을 받은 경우, 어떠한 사유로도 재 채점이 없으므로, 주의할 것.

→ Doc, Hwp 중 하나로 제출 요망

압축 파일 예시:



## [1] 수식 스택 연산 (각 2점씩, 4점)

다음 수식을, array와 linked list를 써서 수업 시간에 배운 2개의 stack을 써서, 각각 (1), (2)를 지시된 데이터구조로 설정한 후 program하시오. 이때, 갈호의 종류를 확인하고, 이를 해결할 수 있는 프로그램을 작성하시오.

A.  $[3 * (20 + 10)] - \{20 / 5 + [4 * 2] / 2\} + 8$ , 답은 90

[2점] 2종의 stack을 Array로 구조를 잡아서 프로그램

B.  $2 * (10 + 15 * 3 - 5) + \{[30 / 3 - (4 / 2) * 3] + 8\} + 6$ , 답은 110

[2점] 2종의 stack을 Linked List로 구조를 잡아서 프로그램

## [2] 수건 돌리기 게임 2종 (6점)

- 무인도에 고립된 20명 (1번~20번 사람들)은 탈출할 수 있는 배의 정원이 4명이라는 현실 앞에 있다. 즉, 모든 사람은 동의 하에 수건돌리기 게임을 하고, 게임의 결과로 탑승자를 선정하기로 하였다. 이를 위해, 눈을 가리고 수건 돌리기를 한다. (이들은 01-Kim, 02-Lee, 03-Choi, 04-Yoon, ..., 15-Park 등이다. 번호와 이름은 프로그래머가 임의로 정하면 된다.)

- 아래 2 게임 모두, 최종 4명이 남을 때까지 수건 돌리기 게임을 계속 한다.

### ■ [1번째 게임, 3점]

- ◆ 처음에 20명의 사람은 자신의 고유 숫자 순서대로 원형의 단 방향 (Single) circle를 형성하고 서 있다.
- ◆ **[주의]** 프로그램 작성시는 반드시 Singly Linked List로 프로그램 해야 한다.
- ◆ 이때, 모두가 보는 가운데 제비 뽑아 임의의 번호를 고른다. 이는, 제일 먼저 수건을 가질 사람을 뽑는 것이다 (이 부분은 Random Number 함수로 프로그램 하면 된다. 예를 들어, 제비를 뽑아, 7번이 뽑힌다면, 7번부터 오른 쪽으로 시작하면 된다.).
- ◆ 그 사람의 번호 즉, 제비로 뽑힌 해당 번호를 포함해서(그 사람부터 시작 해서) 3번째 사람은 그 circle에서 빠진다. 예를 들어, 임의로 뽑았는데 1번-Kim이 뽑혔다고 한다면, 수건을 돌리면, 3번 사람, 03-Choi가 걸리게 되고, circle에서 빠진다.
- ◆ 빠졌으니, 그 수건은 옆 사람인 04-Yoon이 전달받는다. 그리고, 04번부터

동일한 과정을 계속한다. 그 다음 빠질 사람은 06번이다. 이와 같은 과정을 거쳐 점차 인원이 빠져서 원에 최종 4명 만 남을 때까지 이 과정을 반복한다.

#### ■ 문제

- ◆ 처음부터 이러한 과정을 계속 진행하다가, 최종 4인이 남으면, 그들의 번호와 이름을 출력하는 프로그램을 C의 Linked List로 작성하시오.
- ◆ 수건 돌리기 circle에서 빠지는 사람이 발생할 때마다, 빠진 사람의 번호와 이름을 출력하고, 현재 남아 있는 사람들의 List도 한사람이 빠질 때마다, 매번 출력한다. 만약, 16명이 빠지는 경우, 16라인의 프린트가 나올 것이다. 이러한 방법으로 마지막 4인이 남을 때까지 각 단계의 사람들(번호, 이름)을 print하시오.
- ◆ 단방향 수건 돌리기 게임에서 살아 남은 최종 남은 행운의 4명을 답으로 출력 하시오.

#### ■ [2번째 게임, 3점]

- ◆ 1번째 게임과 같이, 처음에 20명의 사람은 자신의 고유 숫자 순서를 가지고 있다. 이번 게임은 “양방향”으로 진행하기로 하였다. 즉, 양 방향(double) circle를 형성하고 서 있다.
- ◆ **[주의]** 프로그램 작성시는 반드시 Doubley Linked List로 프로그램 해야 한다.
- ◆ 이때, 모두가 보는 가운데 제비 뽑아 임의의 번호를 고른다. 이는, 제일 먼저 수건을 가질 사람을 뽑는 것이다 (이 부분은 Random Number 함수로 프로그램 하면 된다, 예를 들어, 제비를 뽑아, 2번이 뽑힌다면, 2번부터 오른쪽으로 시작하면 된다.).
- ◆ 홀수번째 시도에는 오른쪽 시계방향, 짝수번째 시도에는 왼쪽 시계반대 방향으로 게임을 진행하기로 하였다.
- ◆ 즉, 처음에는 랜덤 제비의 번호 즉, 제비로 뽑힌 해당 번호를 포함해서(그 사람부터 시작해서) 오른쪽 시계방향 게임 시에는 3번째 사람이 그 circle에서 빠진다. 예를 들어, 임의로 뽑았는데 1번-Kim이 뽑혔다고 한다면, 오른쪽 시계 방향으로 수건을 돌리면, 3번 사람, 03-Choi가 걸리게 되고, 03이 circle에서 빠진다.
- ◆ 첫번째 03이 빠진 후에는, 오른쪽에서 왼쪽 반시계 방향으로 수건 돌리는 방향이 바뀐다. 즉, 이번에는 짝수번째 수건 돌기기가 되어 돌리는 방향이 왼쪽으로 바뀐 것이다. 그런데, 왼쪽 반시계 방향으로 수건 도는 방향이 바

편 경우에는 8번째 사람이 그 circle에서 빠진다.

- ◆ 예를 들어, 조금 전에 03번이 빠졌으니, 그 수건은 왼쪽 옆 사람인 02-Lee가 전달받는다. 그리고, 02번부터 동일한 과정을 8번 반복한다. 즉, 02, 01, 20, 19, 18, 17, 16, 15, 즉, 그 다음 빠질 사람은 15번이다.
- ◆ 이와 같은 과정을 거쳐 점차 인원이 빠져서 양방향 원에 최종 4명 만 남을 때까지 이 과정을 반복한다.

■ 문제

- ◆ 처음부터 이러한 과정을 계속 진행하다가, 최종 4인이 남으면, 그들의 번호와 이름을 출력하는 프로그램을 C의 Linked List로 작성하시오.
- ◆ 수건 돌리기 circle에서 빠지는 사람이 발생할 때마다, 빠진 사람의 번호와 이름을 출력하고, 현재 남아 있는 사람들의 List도 한사람이 빠질 때마다, 매번 출력한다. 만약, 16명이 빠지는 경우, 16라인의 프린트가 나올 것이다. 이러한 방법으로 마지막 4인이 남을 때까지 각 단계의 사람들(번호, 이름)을 print하시오.
- ◆ 양방향 수건 돌리기 게임에서 살아 남은 최종 남은 행운의 4명을 답으로 출력하시오.

### [3] 연결 리스트 정렬 검색 (각 2점, 6점)

다음 31개의 값 (66, 2, 67, 69, 8, 11, 43, 49, 5, 6, 70, 71, 73, 75, 48, 12, 14, 7, 15, 24, 26, 42, 51, 55, 56, 59, 1, 3, 80, 96, 99)이 키(key)로 주어졌다. 순서대로 받아들여 linked list를 형성한 후, 아래에 지시한 정렬 방식을 사용하여 정렬하고, 검색한다.

- [정렬, 3점] 31개의 수를 입력으로 받아, 이를 선택 정렬(selection sort)로 정렬 하시오.
- [검색, 3점] 수업 시간에 배운 3차시에 학습한 이진 탐색(Binary Search) 프로그램을 구현하시오.
  - [검증 방법] 상기 정렬된 linked list에 다음 10개의 숫자가 존재하는지 확인한다. 검색 결과는 (1)~(10)의 각 항목별로 답을 하시오. 검색 결과,
    - (A) 리스트에 숫자가 존재하면, True, 존재하지 않으면 False를 프린트하고, 연이어,
    - (B) 해당 수가 리스트 내에 존재한다면, 그 숫자가 몇 번째 노드에 존재하는지를 프린트한다. 만약, 리스트 내에 숫자가 없으면 “No Result in List”를 프린트

하는 프로그램을 작성한다.

- 답이 틀리면 감점됨.
- 10가지 검색 문제들: (1) 3, (2) 97, (3) 96, (4) 111, (5) 15, (6) 9, (7) 66, (8) 120, (9) 99, (10) 59

#### [4] 미로 탈출 프로그램 (4점)

- 미로 탐색 문제란 미로에 갇힌 생쥐가 출구를 찾는 문제이다. 미로는 서로 연결된 여러 개의 작은 방 또는 칸으로 구성되어 있다고 가정하자.
- 생쥐가 출구를 찾는 기본적인 방법은 시행착오 방법으로서 하나의 경로를 선정하여 한번 시도해 보고 안되면 다시 다른 경로를 시도하는 것이다.
- 문제는 현재의 경로가 아닐 경우에 다른 경로를 선택해야 한다는 것으로 가능한 경로들이 어딘가에 저장되어 있어야 한다는 것이다. 그러면 어디에 가능한 경로들을 저장하는 것이 좋을까? 현재 위치에서 가능한 경로 중에서 가장 가까운 경로가 저장되면 좋을 것이다. 스택을 사용하면 좋을 듯하다.
- 현재 위치에서 갈 수 있는 좌표들을 스택에 기억하였다가 막다른 길을 만나면 가장 가깝고 아직 가보지 않은 칸으로 다시 돌아가서 새로운 경로를 찾아보는 것이다.
- 또한 한번 지나간 경로 또는 칸을 다시 가서는 안될 것이다. 이렇게 하기 위해서는 한번 지나간 칸에는 이미 방문했음을 알리는 표식을 해 놓으면 될 것이다.
- 이러한 개념을 가지고, 다음에 제시하는 여러 그림들을 참고해서 상상해 보고, 프로그램에서 미로의 크기를 10 x 10으로 설정한 후 프로그램을 수행하여, 미로 문제에서 스택의 사용 전 과정을 빠짐없이 프린트하는 부분을 자세히 추가해 보면서 미로 프로그램을 확인해 보는 문제를 풀어본다.
- 구현 방법은 프로그래머가 array 또는 linked list를 선정하여 임의로 진행 가능하다.
- 만약, 미로 프로그램에서 본인이 선행적으로 설정한 선행 값들이 있다면, 이를 제시하고 자유롭게 프로그램 하는 것을 허락한다.
- 예시 그림은 다음과 같다.



```

스택 S와 출구의 위치, 현재 생쥐의 위치를 초기화
while( 현재의 위치가 출구가 아니면 )
do 현재 위치를 방문한 것으로 표기
  if( 현재 위치의 위, 아래, 왼쪽, 오른쪽 위치가 아직 방문되지 않았고 갈수 있으면 )
  then 그 위치들을 스택 s에 push
  if( is_empty(s) )
  then 실패
  else 스택에서 하나의 위치를 꺼내어 현재 위치로 만든다;
성공;

```

