**데이터구조 및 프로그래밍실습 3분반**

**설계 프로젝트**

학번 : 2020106209

이름 : 김경현

본인의 Github 주소 :

https://github.com/kyeonghyeon0314/DS\_project.git

**문제 1.**

1. **본 문제를 풀기위해 사용한 데이터 구조를 제시하고 간단히 설명하세요.**  
     
   1차원 배열의 안의 값의 합, 배열의 길이, 배열 sorting을 사용하여 평균값과 중앙값을 계산하였습니다.  
   sorting할 때 sort 함수를 사용하였는데 sort 함수는 Timsort라는 알고리즘을 사용한다고 합니다. 저희가 배운 데이터구조 중에서 insertion sort와 merge sort를 결합한 하이브리드 정렬 알고리즘이라고 합니다.   
   Timsort 문서 : <https://en.wikipedia.org/wiki/Timsort>  
   Python 문서 : <https://docs.python.org/3/howto/sorting.html>
2. **텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

   자동 생성된 설명본인이 작성한 파이썬 Code 캡처 이미지를 첨부하고 Algorithm Analysis를 수행하세요.**   
     
     
   평균값(mean)은 sum과 len을 할용하여 구한후 input 배열의 각 요소들의 합과 input 배열의 길이를 / 연산자를 사용하여 mean을 구하였습니다.  
     
   중앙값(median)을 구할 때는 우선 input 배열을 sort를 활용하여 크기순으로 정렬하고 배열의 길이가 짝수 일 때와 홀수 일 때 구분하여 중앙값을 구하였습니다.

**문제 2.**

1. 본 문제를 풀기위해 사용한 데이터 구조를 제시하고 간단히 설명하세요.  
     
   괄호를 열고 닫을 것을 생각했을 때 stack을 활용하여 여는 괄호를 저장해 두었다가 출력 닫는 괄호가 얼마나 필요할지 확인하면 해결 할 수 있을 것이라고 생각하였습니다.|
2. 텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

   자동 생성된 설명본인이 작성한 파이썬 Code 캡처 이미지를 첨부하고 Algorithm Analysis를 수행하세요.   
     
   FOR 문  
   **여는 괄호를 만나면** 스택에 추가하고 출력 문자열에도 추가합니다.  
     
   **닫는 괄호를 만나면** 스택이 비어 있을 때와 비어 있지 않을 때로 구분합니다.  
   ***스택이 비어 있지 않***다면 스택에서 하나를 제거합니다. ***비어 있을 때***는 추가해야 할 여는 괄호 수를 증가시키고 출력 문자열의 앞에는 여는 괄호를 추가합니다.  
   출력 문자열에도 닫는 괄호를 추가합니다.  
     
   WHILE문  
   스택에 남아 있는 여는 괄호의 수를 결과에 더해주고 출력 문자열에는 닫는 괄호를 추가해줍니다.

**문제 3.**

1. 본 문제를 풀기위해 사용한 데이터 구조를 제시하고 간단히 설명하세요.  
     
   BFS를 구현하기 위해 큐를 사용하였습니다. 큐는 deque를 이용해 구현하였으면 큐에 곰의 현재 위치와 이동거리를 저장하고, 이 위치를 기준으로 상하좌우 탐색을 진행합니다. 숲의 상태와, 방문, 먹을 수 있는 벌집의 위치와 거리를 저장하는 리스트를 활용하였습니다. 그리고 문제 1번에서 소개한 sort함수를 활용하였습니다. 그러나 이번의 경우에는 lambda 함수를 활용해 거리, 위, 좌를 기준으로 정렬을 하였습니다.  
   lambda 함수 문서 : <https://docs.python.org/3/tutorial/controlflow.html#lambda-expressions>
2. 본인이 작성한 파이썬 Code 캡처 이미지를 첨부하고 Algorithm Analysis를 수행하세요.   
    **BFS**곰의 현재 위치에서 시작하여 bfs 탐색을 수행합니다. 대기열을 사용하여 방문할 셀과 시작 위치로부터 거리를 추적합니다.   
   방문 배열을 유지하여 셀이 다시 방문하는 것을 방지합니다. 셀에서 곰이 먹을 수 있는 벌집이 포함된 경우 먹을 수 있는 벌집 목록에 추가 됩니다.  
   BFS는 큐가 비어 있을 때까지(도달 가능한 모든 셀이 탐색될 때까지)계속됩니다.  
     
   **MAIN LOOP**곰이 먹을 수 있는 가장 가까운 벌집을 차기 위해 bfs 기능을 반복적으로 호출합니다. 벌집이 발견되면 곰은 위치로 이동하여 벌집을 먹고 시간과 벌집 수가 업데이트됩니다.  
   곰이 현재 크기와 같은 수의 벌접을 먹으면 크기가 1증가합니다.  
   이 루프는 먹을 벌집이 존재하지 않을 때 까지 계속 됩니다.  
     
     
   ps.코드에 대한 설명은 파이썬 파일에 주석으로 달아 놓았습니다.  
     
     
     
     
     
     
   텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

   자동 생성된 설명