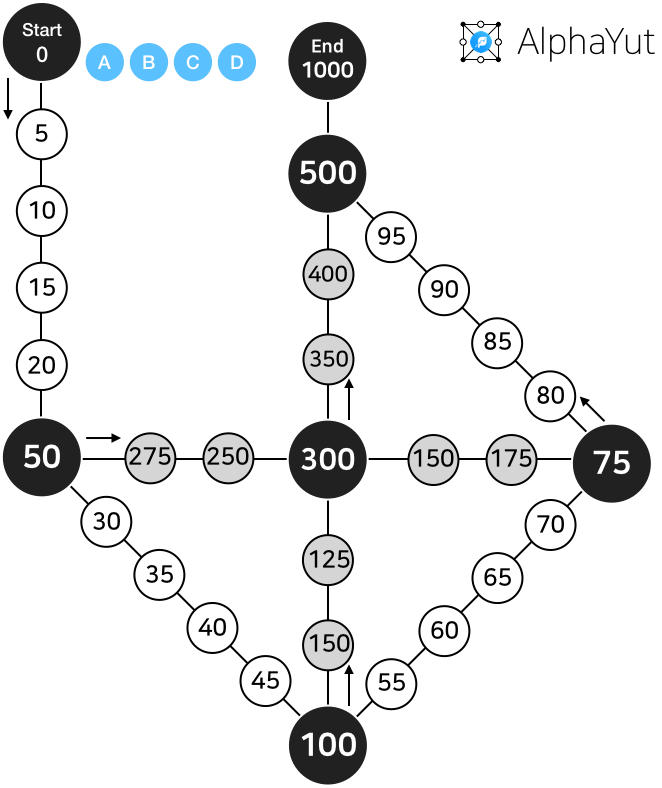
알파윷

이전다음

*시간제한 3초*

**문제**

알파고를 보고 감명받은 한국의 박사들은 윷놀이의 알파고, "알파윷"을 만들기로 했다. "알파윷"은 윷놀이 인공지능으로 윷놀이 게임에서 어떠한 상대를 만나도 이길 수 있도록 만드는 것이 목표다.

  
[그림 1]

알파윷이 윷놀이 판을 볼 때, 윷놀이 판의 각 칸에는 점수가 적혀있다. 칸에 적힌 점수가 높을 수록 최종적으로 이길 확률이 높아진다. 따라서, 알파윷의 차례에 말들을 모두 이동시킨 뒤 최종적으로 말들이 놓인 칸에서 얻은 점수들의 총 합이 매번 최대가 되는 것이 이기기 위한 필수 조건이다. 예를 들어, 이번 차례에 이동을 모두 마친 뒤 각 말이 놓인 칸의 점수가 A 말이 110, B 말이 200, C 말이 500, D 말이 80이라면 890점 정도 이길 확률이 있다는 의미다.

윷을 10번 던지게 되었을때 이동해야되는 칸 수를 미리 알고 있다. 즉, 10번의 상황 각각에서 도, 개, 걸, 윷, 모 중 어떤 것이 나올지 미리 알고 있는 상황이다. 이때, 10번의 이동 상황에서 적절히 말을 선택하여 최대한 많은 점수를 얻어야한다. 이동하는 칸의 수는 도, 개, 걸, 윷, 모 로 각각 1, 2, 3, 4, 5칸을 뜻한다.

알파윷이 말을 이동시킬 때에 2 가지 규칙이 있다.

* 첫째로, 이동시 한 칸에 2마리 이상의 말이 존재할 수 없다. 예를 들어, A 말이 있는 자리에 B, C, D 말이 올 수 없다.
* 둘째로, 검은 칸에 도착할 경우 무조건 화살표 있는 방향으로 이동해야한다.
* 셋째로, 기본적인 이동은 흰 칸이 있는 방향으로 이동해야한다.

윷놀이의 기본 규칙은 다음과 같다.

* 시작 지점에 말이 4개 존재한다.
* 만약, 이동해야하는 칸이 도착 지점을 넘어서거나, 도착 지점을 밟았을 경우 도착 지점의 점수를 얻고 해당 말은 판에서 제외된다.
* 판에서 제외된 말은 더 이상 이동할 수 없다. 또한, 제외된 말에 이동 명령을 내릴 수 없다.

말 선택 알고리즘에 치명적인 버그가 발생하여, 새로 만들어야하는 상황이다. 현재 윷놀이 판의 점수 정보와 각 라운드별 이동 칸 수가 주어졌을때, 말 4개를 적절히 선택하여 이동을 마친 후 말들이 존재하는 칸의 점수의 합이 최대가 되도록 만들어보자.

**입력**

첫 번째 줄에 테스트 케이스의 개수 T(=50)가 주어진다. 다음 줄부터 T개의 테스트 케이스에 대한 정보가 주어진다. 각각의 테스트 케이스는 한 줄로 이루어져있고, 10개의 숫자가 공백을 통해 구분하여 주어진다. 각각의 숫자는 1부터 5까지의 자연수이다.

**출력**

각 테스트 케이스에 대해 만들 수 있는 숫자 합 중 최대값을 출력한다. 각 테스트 케이스의 출력 양식은 "#t r"이다. t는 테스트 케이스의 번호이며, 1부터 시작한다. r은 문제에 대한 결과값을 뜻한다.

**예제 입력**

copy

5

1 2 4 3 2 5 2 1 5 3

2 3 5 1 3 3 1 2 5 1

2 5 5 5 3 4 1 5 4 3

5 4 4 4 2 5 2 3 4 3

5 3 4 5 3 4 5 3 4 5

**예제 출력**

copy

#1 2010

#2 1525

#3 2065

#4 2035

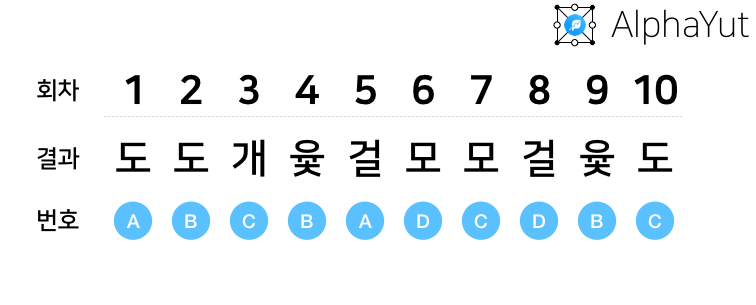
#5 3050

하위모듈

# 조합 알고리즘

이전다음

### 문제



현재 윷놀이 판에 말이 A, B, C, D로 총 4개 존재한다. 10번 윷을 던져 나온 결과를 미리 알고 있다. 결과는 도, 개, 걸, 윷, 모로 1, 2, 3, 4, 5 칸 이동하게 된다. 각각의 결과를 말 4개에 적절히 분배할 것이다. 분배한 결과를 적을때 편의상 말의 번호를 알파벳 대신 숫자로 표현한다. 다시 말해, A는 1, B는 2, C는 3, D는 4 로 바꿔적는다.

재귀호출을 이용하여 윷을 분배하는 모든 경우의 수를 확인함으로써 문제를 해결하시오. 즉, 1 1 1 1 1 1 1 1 1 1 부터 4 4 4 4 4 4 4 4 4 4 까지 모든 경우를 재귀호출을 활용하여 나열하고, 그 중 K번째 경우를 출력하는 프로그램을 작성하시오.

### 입력

첫 번째 줄에 자연수 K가 주어진다. (1 ≤ K ≤ 1,048,576)

### 출력

한 줄에 K번째 경우에 대해 출력한다. 각 숫자의 사이에는 공백이 들어간다.

### 입력 예시 1

copy

1

### 출력 예시 1

copy

1 1 1 1 1 1 1 1 1 1

### 입력 예시 2

copy

16

### 출력 예시 2

copy

1 1 1 1 1 1 1 1 4 4

### 입력 예시 3

copy

438284

### 출력 예시 3

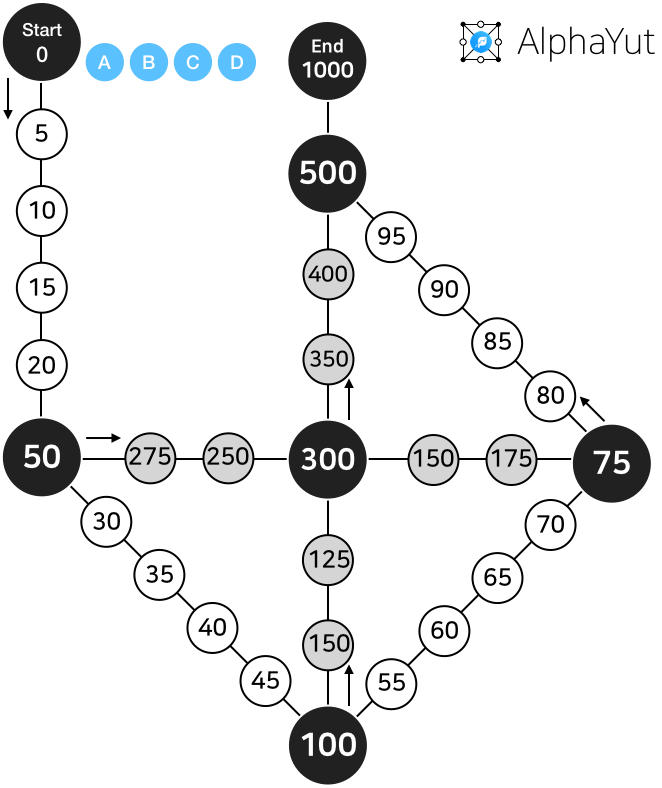
copy

2 3 3 4 1 1 1 1 3 4

# 이동 알고리즘 4

이전다음

### 문제



T개의 윷을 던진 상황이 주어질때, 각 상황에서 얻어지는 점수 중 최댓값을 알아보자.

각 상황에서 말 4개가 움직인다. 얻은 점수의 합이란, 각 말이 얻은 점수의 총 합을 의미하며 각 말은 마지막에 밟았던 칸의 점수를 자신의 점수로 갖는다.

말의 이동 시작점은 Start 이며, 도착지점은 End 이다. 만약, 이동해야하는 칸이 도착 지점을 넘어서거나, 도착 지점을 밟았을 경우 도착 지점의 점수를 얻고 해당 말은 판에서 제외된다. 제외된 말은 더 이상 이동할 수 없다. 50점과 75점 칸을 밟을 경우, 화살표가 가리키는 방향으로 움직여야한다. 기본적으로 흰색 칸이 있는 경로로 이동한다.

이동하려고 하는 칸에 다른 말이 있는 경우 이동할 수 없다. 해당 상황의 결과값은 "-1"이다.

### 입력

첫 번째 줄에 상황의 개수 T가 주어진다. 두 번째 줄부터 T x 2개의 줄에 걸쳐 윷을 던진 상황이 주어진다. 각 상황의 첫 번째 줄에 이동하게될 칸의 수가 10개 주어진다. 그리고 두 번째 줄에 각 이동하게될 칸의 수가 몇 번 말이 될지 주어진다.(2 ≤ T ≤ 10, 1 ≤ 이동하는 칸의 수 ≤ 5, 1 ≤ 말의 번호 ≤ 4)

### 출력

T개의 상황에서 얻어지는 점수 중 최댓값을 알아보자. 만약 주어진 모든 상황에서 얻은 결과값이 -1이라면 "-1"을 출력한다.

### 입력의 예

copy

4

2 3 1 2 4 5 2 3 4 1

1 1 2 3 4 1 2 2 3 1

5 5 5 5 5 5 5 5 5 5

1 2 3 4 1 2 3 4 1 2

5 3 4 5 3 4 5 3 4 5

1 1 1 2 2 2 3 3 3 4

3 4 3 2 1 2 3 4 3 2

1 2 2 3 4 3 1 4 4 1

### 출력의 예

copy

3050