





C. 다트(Hard C)

 문제 설명

 문제 풀이

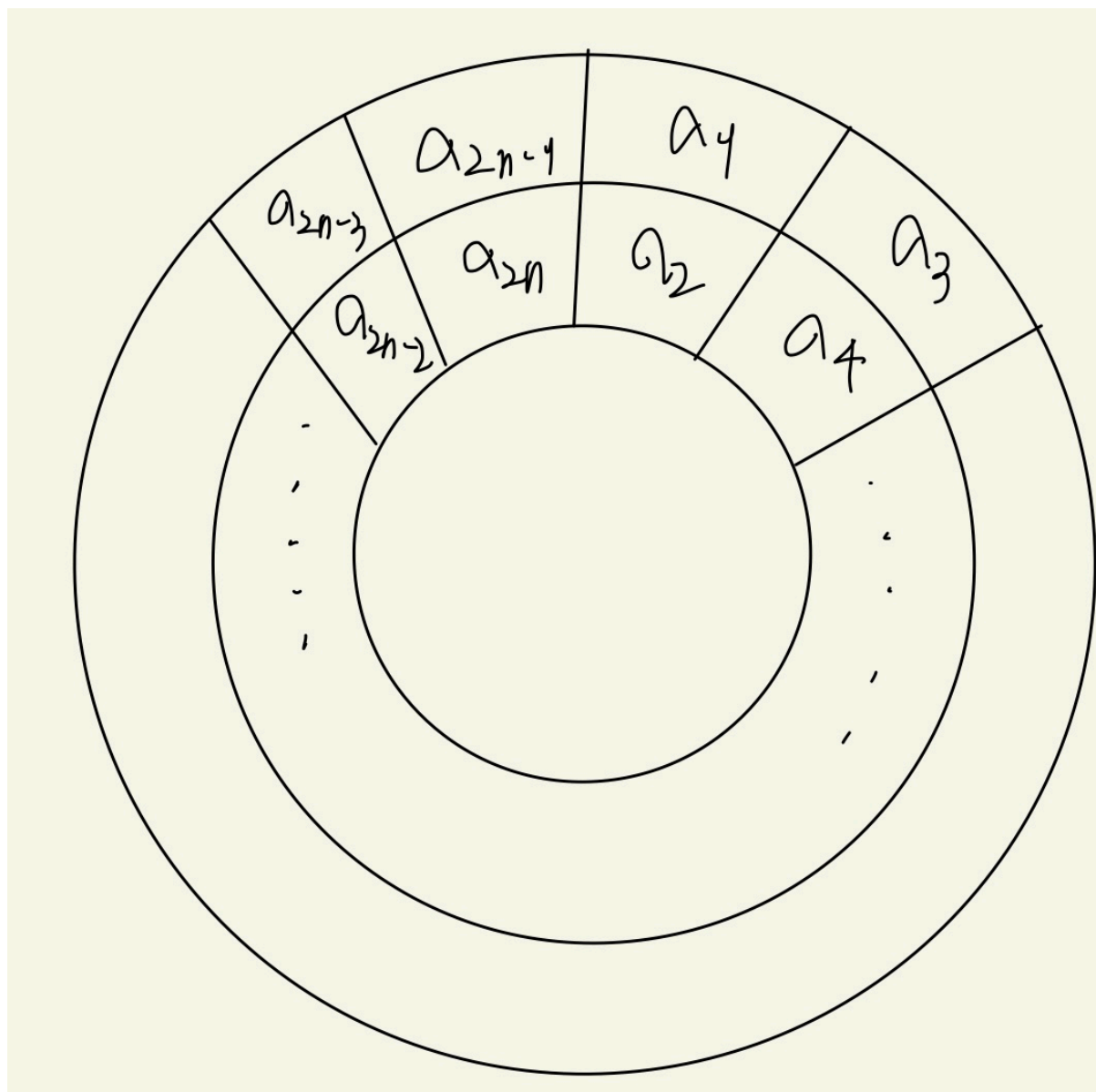
 정해 코드



문제 설명

산지니는 다트를 좋아한다.

산지니가 하고 있는 다트의 다트판은 $2N$ 개의 판으로 이루어져 있으며, 다트판 이렇게 생겼다.



각 다트판의 칸에는 숫자가 적혀있고, 만약 어떤 칸을 맞춘다면 그 칸에 적혀있는 숫자만큼 점수를 얻는다.

하지만, 이 다트 판의 룰이 하나 더 있었다.

- 산지니가 다트를 어떤 칸에 맞추었을 때, 만약 그 칸에서 거리가 k 안인 칸 안에 다트를 맞춘 적이 있을 경우 그 게임은 0점을 받는다.

산지니는 다트를 엄청나게 연습해왔기에 다트를 모든 칸에 정확히 맞출 수 있다.

산지니가 얻을 수 있는 최대의 점수를 구하자!

입출력 조건

입출력 조건

입력

첫 번째 줄에 다트판 한 줄의 길이 $N(1 \leq N \leq 2000)$, $K(1 \leq k \leq N)$ 가 주어진다.

두 번째 줄에 다트판의 칸에 적혀 있는 점수들이 $2 * N$ 개로 주어지며, i 번째 숫자 a_i 는 i 번째 칸의 점수이다.

$(1 \leq a_i \leq 4000)$

출력

산지니가 얻을 수 있는 최대의 점수를 출력한다.



문제 풀이

사용해야 하는 알고리즘

- *Dynamic Programming*

확정 시간/공간 복잡도

- 시간 복잡도: $O(n^2)$
- 공간 복잡도: $O(n^2)$

풀이

일단 두 배열을 설정하자.

- $DP_{i,j,x,y} =$ (현재 위치가 i 번째 x 칸이고, 처음 시작한 위치가 j 번째 y 칸일 때, 최대 얻을 수 있는 점수)
- $MX_{i,j,x,y} = \text{MAX}(DP_{k,j,x,y}) (1 \leq k \leq i, 0 \leq t \leq 1)$

라고 한다면,

일단 조건에 맞지 않는 경우는 자명하게 구할 수 있다. 이때는 그냥 -1을 넣어주자.

점화식은 잘 생각해보면 이렇다.

- $DP_{i,j,x,y} = \text{MAX}(MX_{i-k,j,x,y}, MX_{i-k+1,j,nx,y}) + a_{i,x}$
- $nx = (x + 1) \bmod 2$