Card Game

Problem by: Alex Tung

Data by: Jeffery Hui

Problem Statement

For any permutation P, define A_{max} as the length of longest increasing subsequence (LIS) and B_{max} as the length of the longest decreasing subsequence (LDS).

Given N and K, find a permutation P containing numbers from 1 to N such that $\max(A_{max}, B_{max}) = K$.

Example :

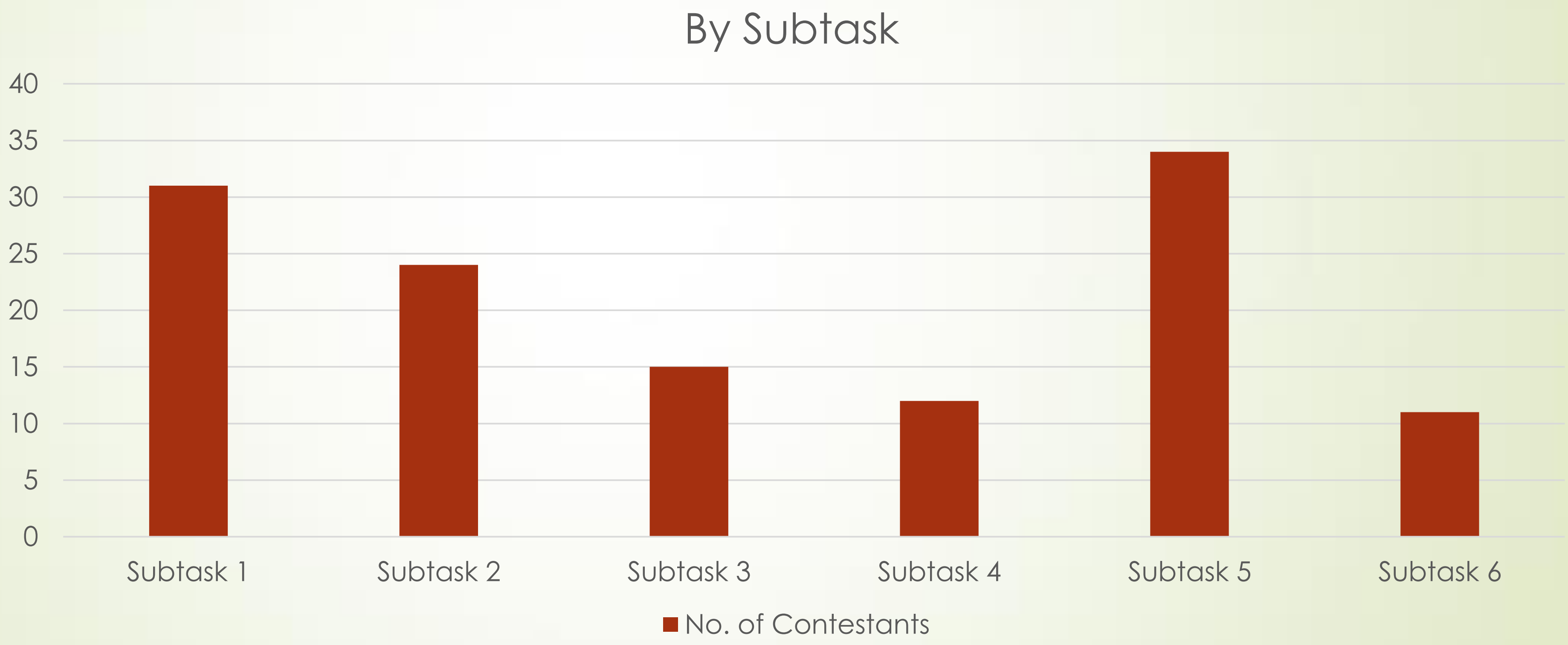
Suppose the permutation is (1, 3, 5, 7, 6, 4, 2), $A_{max} = 4$ (1, 3, 5, 6 or 7), $B_{max} = 4$ (7, 6, 4, 2), so $\max(A_{max}, B_{max}) = 4$.

Subtask

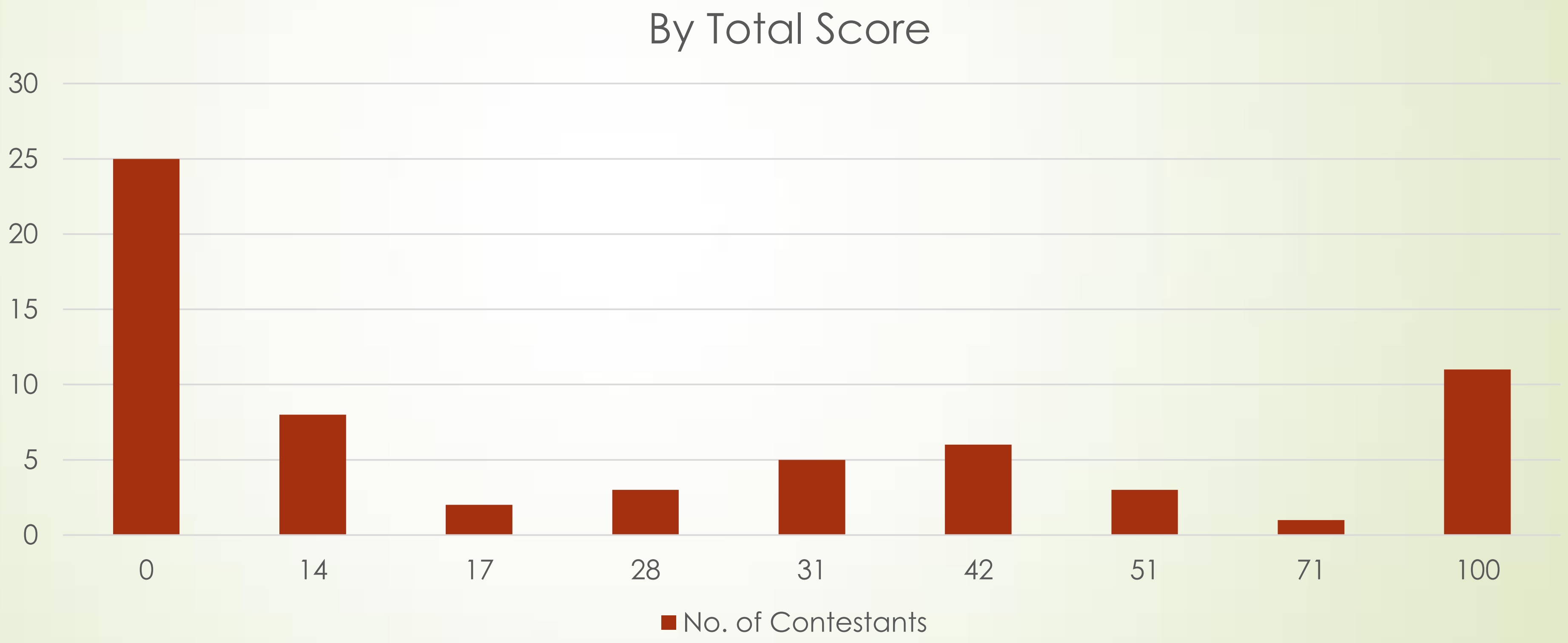
1 <= K <= N <= 100000

- Subtask 1 (17 points): 1 <= N <= 5
- Subtask 2 (11 points): 1 <= N <= 7
- Subtask 3 (9 points): 1 <= N <= 9
- Subtask 4 (20 points): 1 <= N <= 4000
- Subtask 5 (14 points): K > N / 2
- Subtask 6 (29 points): No additional constraints

Score Distribution



Score Distribution



Observation I

- Try some larger K, the answer will be trivial.
- Since there are at most one comment element in LIS and LDS.
  - Proof: If there were more than one comment element in LIS and LDS, it would mean there exist some sequences which would be increasing and decreasing at the same time which is impossible.
- So, $A_{max} + B_{max} <= N + 1$
- Which means
  - if $A_{max} > N/2$
    - then $B_{max} <= A_{max}$
  - If $B_{max} > N/2$
    - then $A_{max} <= B_{max}$

Solution I

- If K > N/2, we can make either A_{max} or B_{max} to be K and this will be the solution.

```
if K > N/2
    for i = 1 to K – 1
        print i
    for i = N down to K
        print i
```

Solution I

- Expected score: 14
  - Subtask 1 **Wrong Answer**
  - Subtask 2 **Wrong Answer**
  - Subtask 3 **Wrong Answer**
  - Subtask 4 **Wrong Answer**
  - Subtask 5 **Accepted**
  - Subtask 6 **Wrong Answer**

Exhaustion I

- For small N, we can exhaust all permutation and check all the subsequence to find a solution
  - Time complexity : $O(n!2^n)$
- Expected score: 42
  - Subtask 1 **Accepted**
  - Subtask 2 **Accepted**
  - Subtask 3 **Wrong Answer**
  - Subtask 4 **Wrong Answer**
  - Subtask 5 **Accepted**
  - Subtask 6 **Wrong Answer**

Exhaustion II

- For those who have more advanced knowledge, you may know some other algorithm such as DP and greedy + binary search to find LIS which can improve the solution to a time complexity of $O(n!n^2)$ or $O(n!\ n \lg n)$.
- Expected score: 51
  - Subtask 1 **Accepted**
  - Subtask 2 **Accepted**
  - Subtask 3 **Accepted**
  - Subtask 4 **Wrong Answer**
  - Subtask 5 **Accepted**
  - Subtask 6 **Wrong Answer**

Hardcode

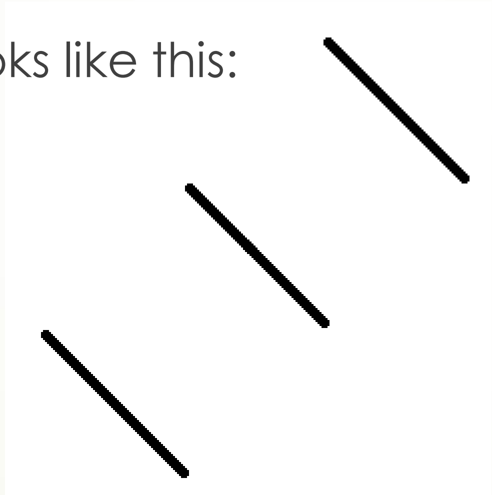
- Instead of exhaustion, we can hardcode for small N.
- Since we have solved for K > N/2, we can consider K <= N/2 only.
- For N <= 5 , almost all cases are impossible except for N = 4, K = 2, one of the solutions is 2 1 4 3.
- For 5 < N <= 7, the exceptional cases are
  - N = 6, K = 3, one of the solutions is 3 2 1 6 5 4
  - N = 7, K = 3, one of the solutions is 3 2 1 6 5 4 7
- For 7 < N <= 9, the exceptional cases are
  - N = 8, K = 3, one of the solutions is 3 2 1 6 5 4 8 7
  - N = 8, K = 4, one of the solutions is 4 3 2 1 8 7 6 5
  - N = 9, K = 3, one of the solutions is 3 2 1 6 5 4 9 8 7
  - N = 9, K = 4, one of the solutions is 4 3 2 1 8 7 6 5 9

Hardcode

- Expected score: 51
  - Subtask 1 **Accepted**
  - Subtask 2 **Accepted**
  - Subtask 3 **Accepted**
  - Subtask 4 **Wrong Answer**
  - Subtask 5 **Accepted**
  - Subtask 6 **Wrong Answer**

Observation II

- After trying some small N, you may find some pattern in the solution.
- Eg. K, K – 1, K – 2, …, 1, 2K, 2K – 1, …, K + 1, 3K, …
- The Solution looks like this:



Solution II

- We divide the permutation into ceil(N/K) parts and each parts contain K numbers
- Then reverse the order in every parts.
- We know A_{max} = ceil(N/K) and B_{max} = K

- This will give us a solution iff $A_{max} <= B_{max}$, ie. $K^2 >= N$
- Actually, there will be no solution for $K^2 < N$ (can use Erdős–Szekeres theorem)
- Alternatively, we can prove max(Amax, Bmax) >= sqrt(n)

Solution II

Proof:

- Let the permutation be $x_1, x_2, x_3, \ldots, x_n$
- Let LIS[i] be the length of longest increasing subsequence in which x_i is the last element
- Also, define LDS[i] in the similar way.
- We know that (LIS[i], LDS[i]) is pairwise distinct distinct for all i = 1, 2, …, n
  - for any i < j, if LIS[i] = LIS[j] (=> $x_i > x_j$ otherwise LIS[i] < LIS[j]), then LDS[i] < LDS[j]
- By pigeonhole principle, max(LIS[i]) × max(LDS[i]) >= n
- Which means $A_{max} \times B_{max}$ >= n
- So, there will be no solution for $K^2 < N$ (=> $A_{max} \times B_{max} < n$)

Solution II

```
if (K * K < N)
    print Impossible
else
    for i = 1 to N/K
        for j = i * K down to (i - 1) * K + 1
            print j
    if (N % K != 0)
        for j = N down to N/K * K + 1
```

Solution II

- Expected score: 100 ☺
  - Subtask 1 **Accepted**
  - Subtask 2 **Accepted**
  - Subtask 3 **Accepted**
  - Subtask 4 **Accepted**
  - Subtask 5 **Wrong Answer** **!!!!!!**
  - Subtask 6 **Wrong Answer** **???**

Solution II

```
if (K * K < N)
    print Impossible
else
    for i = 1 to N/K
        for j = i * K down to (i - 1) * K + 1
            print j
    if (N % K != 0)
        for j = N down to N/K * K + 1
```

Solution II

```
if (K * K < N)
    print Impossible
else
    for i = 1 to N/K
        for j = i * K down to (i - 1) * K + 1
            print j
    if (N % K != 0)
        for j = N down to N/K * K + 1
```

Solution II++

```
if ((long long)K * K < N)
    print Impossible
else
    for i = 1 to N/K
        for j = i * K down to (i - 1) * K + 1
            print j
    if (N % K != 0)
        for j = N down to N/K * K + 1
```

Solution II++

- Expected score: Real $\cdot 100$
  - Subtask 1 **Accepted**
  - Subtask 2 **Accepted**
  - Subtask 3 **Accepted**
  - Subtask 4 **Accepted**
  - Subtask 5 **Accepted**
  - Subtask 6 **Accepted**

Q&A