1 Sorting Algorithm

1.1 Patient Sorting

Consider the following game: deal cards $c_1, c_2, ..., c_n$ into piles according to two rules

- Can't place a higher valued card onto a lower valued card;
- Can form a new pile and put a card onto it.

And the goal is to form as few piles as possible.

Greedy Algorithm In the natural order, place each card on the leftmost pile that fits. Note that by construction, at any stage during the greedy algorithm, top cards of piles increase from left to right.

1.1.1 Connection with LIS

Lemma 1.1 (Weak Duality)**.** *In any legal game of the patience, the number of piles is larger than or equal to any increasing subsequence.*

Proof. Note that cards within a pile form a decreasing subsequence and any sequence can use at most one card from each pile. □

Lemma 1.2 (Strong Duality)**.** *Min number of piles is equal to LIST. More over, the greedy algorithm finds both.*

Proof. Each card maintains a pointer to top card in previous pile. The we can follow pointers to obtain IS whose length is equal to the number of piles. The by the weak duality lemma above, the sequence if one member of the optimal solution set of the LIS problem. □

Implementations This patient sorting algorithm could be implemented in $O(n \log n)$ running time by using an array of stack to represent the piles and binary search to bind the left most pile.

2 Elementary Data Structures

2.1 Monotone Stack

2.1.1 Find k -element subsequence that is lexicographically largest

2.1.2 Find [previous/next] [greater/smaller] element

3 String Algorithms

3.1 Prefix Function

Given a string s of size n, the prefix function $\pi(i) : [0, ..., n-1]$ is defined as

$$\pi[i] = \max_{k=0,...,i} \{k : s[0 : k-1] = s[i-(k-1) : i]. \tag{3.1}$$

In words, it represents the length of the longest prefix substring in $s[0 : i]$ that is also a suffix.