Longest Increasing Subsequence

Longest Increasing Sequence

Longest increasing sequence. Given a sequence of elements $c_1, c_2, \ldots, c_n$ from a totally ordered set, find the longest increasing subsequence.

Ex: 7 2 8 1 3 4 10 6 9 5

Maximum Unique Match finder

Application. Part of MUMmer system for aligning entire genomes.

Dynamic programming solution. $O(n^2)$.

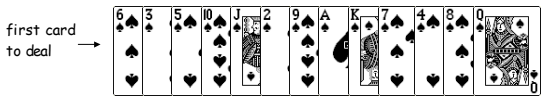
- LIS is a special case of edit-distance.
  - $x = c_1\ c_2\ \ldots\ c_n$
  - y = sorted sequence of c_i, removing any duplicates
  - mismatch penalty = ∞

Patience

Patience. Deal cards $c_1, c_2, \ldots, c_n$ into piles according to two rules:

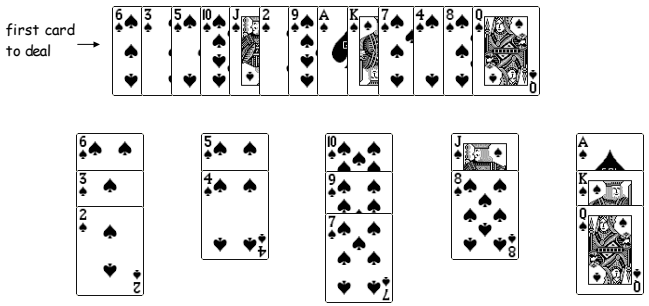
- Can't place a higher valued card onto a lowered valued card.
- Can form a new pile and put a card onto it.

Goal. Form as few piles as possible.



Patience: Greedy Algorithm

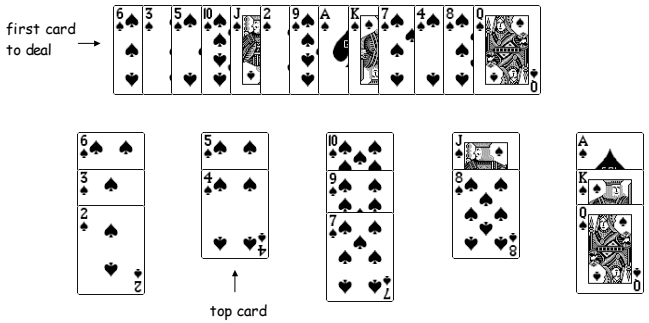
Greedy algorithm. Place each card on leftmost pile that is legally possible to use.



Patience: Greedy Algorithm

Greedy algorithm. Place each card on leftmost pile that is legally possible to use.

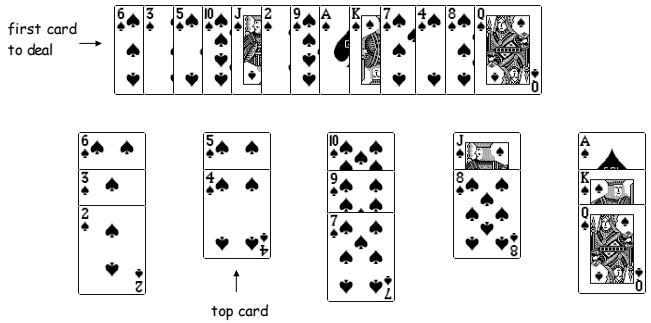
Observation. At any stage during greedy algorithm, top cards of piles increase from left to right.



Patience-LIS: Weak Duality

Weak duality. In any legal game of patience, the number of piles ≥ length of any increasing subsequence.

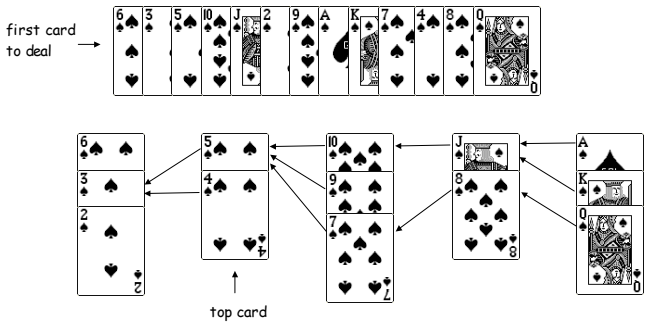
Pf. Cards within a pile form a non-increasing subsequence. An increasing sequence can use at most one card from each pile. ▪



Patience-LIS: Strong Duality

Strong duality. The minimum number of piles = maximum length of an increasing subsequence. Moreover, greedy algorithm finds both.

Pf. Each card maintains a pointer to top card in previous pile (at time of insertion). Follow pointers to obtain an increasing subsequence whose length equals the number of piles. By weak duality, both are optimal. ▪



Greedy Algorithm: Implementation

Efficient implementation. O(n log n)

- Use n stacks to represent n piles.
- Use binary search to find leftmost legal pile.

```
Create n empty stacks S_1, …, S_n

foreach (card c in deck order) {
   S_j ← binary search to find leftmost stack that fits c
   put c on top of stack S_j
   make card c point to top card of stack S_j-1

}

Form LIS by following back-pointers from top card of
rightmost nonempty stack
```

Patience Sorting

Patience sorting. Deal all cards; repeatedly remove smallest card.

Theorem. Expected number of piles is approximately $2n^{1/2}$ with standard deviation around $n^{1/6}$ if deck is uniformly random.

Remark. An almost-trivial $O(n^{3/2})$ sorting algorithm.

Speculation. [Persi Diaconis] Patience sorting is the fastest way to sort a pile of cards by hand.

Bonus Theorem

Theorem. [Erdös-Szekeres, 1935] A sequence of $n^2 + 1$ distinct real numbers either has an increasing or decreasing subsequence of size $n + 1$.

Pf. (by pigeonhole principle)