

1. For any positive integer n , the n th **Fibonacci string** F_n is defined recursively as follows, where $x \cdot y$ denotes the concatenation of strings x and y :

$$F_1 := 0$$

$$F_2 := 1$$

$$F_n := F_{n-1} \cdot F_{n-2} \quad \text{for all } n \geq 3$$

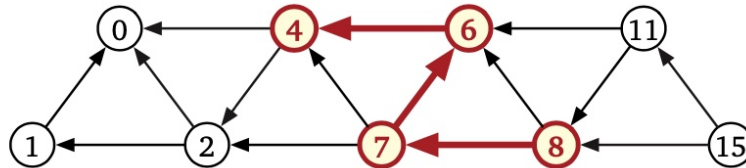
For example, $F_3 = 10$ and $F_4 = 101$.

- (a) What is F_8 ?
- (b) **Prove** that every Fibonacci string except F_1 starts with **1**.
- (c) **Prove** that no Fibonacci string contains the substring **00**.

2. Let G be a **directed** graph, where every vertex v has an associated height $h(v)$, and for every edge $u \rightarrow v$ we have the inequality $h(u) > h(v)$. Assume all heights are distinct. The *span* of a path from u to v is the height difference $h(u) - h(v)$.

Describe and analyze an algorithm to find the **minimum span** of a path in G with **at least** k edges. Your input consists of the graph G , the vertex heights $h(\cdot)$, and the integer k . Report the running time of your algorithm as a function of V , E , and k .

For example, given the following labeled graph and the integer $k = 3$ as input, your algorithm should return the integer 4, which is the span of the path $8 \rightarrow 7 \rightarrow 6 \rightarrow 4$.



1. For any two sets X and Y of integers, the Minkowski sum $X + Y$ is the set of all pairwise sums $\{x + y \mid x \in X, y \in Y\}$.
- (a) Describe and analyze an algorithm to compute the number of elements in $X + Y$ in $O(n^2 \log n)$ time, where $n = |X| + |Y|$. [Hint: The answer is **not** always $|X| \cdot |Y|$.]
- (b) Describe and analyze an algorithm to compute the number of elements in $X + Y$ in $O(M \log M)$ time, where M is the largest absolute value of any element of $X \cup Y$.

3. Suppose you are given two sorted arrays $A[1..n]$ and $B[1..n]$ containing distinct integers. Describe a fast algorithm to find the median (meaning the n th smallest element) of the union $A \cup B$. For example, given the input

$$A[1..8] = [0, 1, 6, 9, 12, 13, 18, 20] \quad B[1..8] = [2, 4, 5, 8, 17, 19, 21, 23]$$

your algorithm should return the integer 9. *[Hint: What can you learn by comparing one element of A with one element of B?]*

2. You and your eight-year-old nephew Elmo decide to play a simple card game. At the beginning of the game, the cards are dealt face up in a long row. Each card is worth a different number of points. After all the cards are dealt, you and Elmo take turns removing either the leftmost or rightmost card from the row, until all the cards are gone. At each turn, you can decide which of the two cards to take. The winner of the game is the player that has collected the most points when the game ends.

Having never taken an algorithms class, Elmo follows the obvious greedy strategy—when it's his turn, Elmo *always* takes the card with the higher point value. Your task is to find a strategy that will beat Elmo whenever possible. (It might seem mean to beat up on a little kid like this, but Elmo absolutely *hates* it when grown-ups let him win.)

- (a) Prove that you should not also use the greedy strategy. That is, show that there is a game that you can win, but only if you do *not* follow the same greedy strategy as Elmo.
- (b) Describe and analyze an algorithm to determine, given the initial sequence of cards, the maximum number of points that you can collect playing against Elmo.