CSE 431/531 Analysis of Algorithms
Problem Set 2

Chen Xu

Due Date: June 29, 2023 23:59 PM EST.

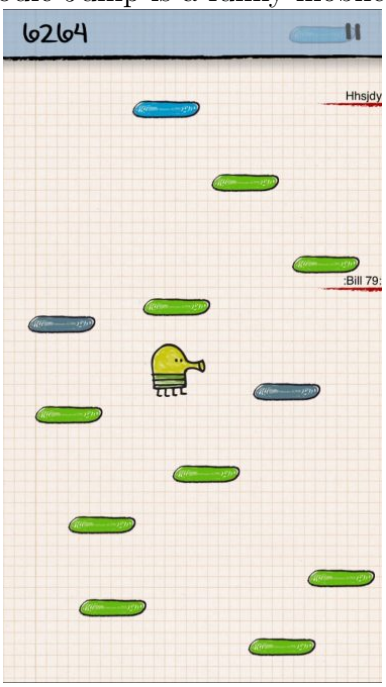
Problem 1. (20%)
A *balanced cut* of a connected tree $T = (V, E)$ (Connected means every node is reachable from every other node) is an edge that removing it cuts the tree into two trees whose vertex sets are V_1, V_2 such that $||V_1| - |V_2||$ is minimized.

1. Show that the balanced cut may not be unique for a tree.
2. Given a connected tree of n vertices, design an efficient algorithm to find one of the balanced cut. Prove the correctness.

Problem 2. (20%)
A *joint* is a 3-tuple of vertices (v_i, v_j, v_k) such that $(v_i, v_j), (v_j, v_k) \in E$. A graph is *joint matched* if there exists a set of joints such that each vertex is present exclusively in one of the joints. In other words, there is a set of "disjoint joints" that covers the graph.

1. Give an example of graph that is joint matched.
2. Give an example of graph that is not joint matched.
3. Given a tree $T = (V, E)$ of n vertices, design an efficient algorithm to determine if it can be joint matched. Prove the correctness.

Problem 3. (20%)
Doodle Jump is a funny mobile game. Here is a screenshot of the game:



The player can control the left right movement to pick where to land. The little creature will bounce on platforms to reach higher places. When it lands on the green platform, it will bounce up a certain height h and be able to reach any platform or objects within that height.

1. Suppose the green platforms' heights were given in a sorted array H. Design an algorithm to pick a subset of platforms to land in order to minimize the number of bounces to reach the top platform. State the induction of subproblems and the solution mapping of your approach. Prove that your algorithm gives optimal solution.

2. Now we consider adding springs. When you land on springs, it will bounce you up $2h$. The spring platforms are also given in a sorted array S. Together with H, design an algorithm to minimize the number of bounces to the top. State the induction of subproblems and the solution mapping of your approach. Prove that your algorithm gives optimal solution if you can.

Problem 4. (20%)
A subword of a string is a string induced by removing any characters from the original string. For instance, 025 is a subword of 021354 by removing 1,3,4. Now let us consider the binary strings of which the characters can only be 0,1. We are interested in the number of occurrences of a subword in a binary string. For example, the number of occurrences of 01 in 0011 is 4 since you can remove either one of the two 0s and 1s to get 01. There are 2×2 possible ways.

1. What is the number of occurrences of 101 in 100101?
2. Given the binary subword of length m and the binary string of length n, design an algorithm using Divide and Conquer strategy to compute the number of occurrences. Prove that your algorithm works and give the running time.

Problem 5. (20%)
We were given a $m \times n$ binary matrix. A submatrix of a matrix is a matrix induced by removing any number of rows and columns from the original matrix. We want to know if there exist a 2×2 submatrix of $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ in the original matrix. For simplicity, let us call this 2×2 matrix *Diagonal matrix*.

1. Give an instance that contains a diagonal submatrix.
2. Give an instance that does not contain a diagonal submatrix.
3. Design a Divide and Conquer algorithm to determine if the given $m \times n$ binary matrix has a diagonal submatrix. Prove the correctness and show the running time.