1 Largest Number with an Upper Bound

Give an sorted sequence of numbers, find the largest number that is not greater than an upper bound. For instance, given an array \{1, 3, 8, 10, 15, 20, 35\}, the largest number that is not greater than 25 is 15. Design a divide and conquer algorithm that has complexity $O(\log(n))$, where $n$ is the length of the array.

2 Max-Sum

Given an sequence of numbers, we want to find the sub-sequence that has maximum sum. We require that the sub-sequence be a continuous chunk. A sub-sequence is thus determined by a start position and an end position in the original sequence. For instance, given a sequence \{-1, 2, 4, -7, 9, -1, 10, -5\}, \{-2, 4, -7, 9\} is a feasible sub-sequence. The max-sum sub-sequence for this example is \{9, -1, 10\} whose sum is 18. A naive solution to this problem has a complexity of $O(n^2)$, where $n$ is the length of the sequence. Use divide and conquer to devise a more efficient algorithm. Write its pseudocode and analyze the complexity.

3 Graph 2-Coloring

We want to test whether an undirected graph can be colored with 2 colors. The constraint is that each two vertices linked by an edge must have different colors. If the coloring is possible, your algorithm should give a coloring result; if not, the algorithm should report it. Construct an efficient algorithm to solve this problem. Write the pseudocode.

A related problem to graph 2-coloring is to test whether an undirected graph is bipartite, i.e., it’s vertices can be partitioned into two disjoint sets and the edges only connect vertices between the two sets (there are no edges that connect vertices within each set). Show how to test whether a graph is bipartite by using your graph 2-coloring algorithm.
4 Safest Path

We are on a hiking trail and suddenly the storm is coming; we want to get to a safe point as soon as possible. Now we need to do some planning. We have a map and we identify a set of safe locations (they are equally safe). We know the probability to safely pass each segment of the trail in order to reach these points. For instance, to reach point $A$ from our current location, we need to go through a sequence of trail segments with probabilities $p_1, p_2, \ldots, p_k$. To safely reach $A$, our overall chance is the product of probabilities $p_1 \cdot p_2 \cdot \ldots \cdot p_k$. Design an efficient algorithm to find out the safest path from our current location to some safe point.

5 Diameter of a Tree

The diameter of a tree $T = (V, E)$ is given by $\max_{u,v \in V} d(u,v)$, where $d(u,v)$ is the shortest distance between $u$ and $v$. Design an efficient algorithm to compute the diameter of a tree. Give its complexity.

6 Smart Purchase

Items in a store are on sale. If you buy something, you can take something with a smaller value for free. You have a shopping list $a_1, a_2, \ldots, a_n$ and you want to spend minimum money on these items. Design a greedy algorithm for this problem and discuss whether it always achieves the global optimum.