Greedy Algorithms, Divide and Conquer

1 Greedy Algorithms

Problem 1. FRACTIONAL KNAPSACK
Input: N objects of total values w_i and volumes v_i, and a knapsack of capacity C
Output: The maximum total value of objects put in the knapsack so that total amount of objects in knapsack does not exceed C (a fraction of object is allowed to be selected)
Observation: An object with largest per-unit-volume value w_i/v_i should be taken as much as possible

Problem 2. ACTIVITY SELECTION [2014]
Input: N intervals $\{[a_i, b_i]\}$ representing activities
Output: The maximum number of activities selected so that they do not overlap with each other
Observation: There exists an optimal choice choosing an activity with the earliest end time

Problem 3. WATERING GRASS
Input: N intervals $\{[a_i, b_i]\}$ representing areas covered by sprinklers, and an interval $[a, b]$ representing the area of grass
Output: The minimum number of sprinklers selected so that they cover all grass

Problem 4. ADVERTISEMENT
Input: N intervals $\{[a_i, b_i]\}$ representing routes of joggers
Output: Choose the minimum number of points $\{p_j\}$ to post advertisements so that every jogger sees at least one advertisement

Problem 5. COMPETITION I [2053]
Input: $2N$ numbers $\{a_i\}$ and $\{b_i\}$ representing strengths of two teams of N students which are taking part in N rounds of one-on-one competition
Output: The maximum number of rounds won by the first team
Observation: A student in the first team should compete with the one just weaker than him/her

Problem 6. COMPETITION II [1081]
Input: Same as the previous problem
Output: The maximum number of rounds won by the first team minus that won by the second

Problem 7. PROGRAM LOADER [1089]
Input: An integer K representing the maximum number of programs allowed in memory and a sequence $\{p_i\}$ of N programs to be executed
Output: The minimum number of times a program is loaded into the memory

Problem 8. BRIDGE
Input: The time $\{t_i\}$ it takes for N people to cross a bridge individually
Output: The time required for all of them to cross the bridge, so that at most two cross it at a time, and every time the people crossing the bridge must be holding the only lamp shared by all people
Observation: We should pair up the two fastest people with the two slowest people

Problem 9. ADDITION [2025]
Input: N numbers $\{a_i\}$
Output: The time required for finding their sum, subject to the constraint that two numbers are added at a time and the time it takes to add two numbers a_i and a_j is $a_i + a_j$
Observation: The two smallest numbers should be added first

Problem 10. TRANSPORT I
Input: N piles of objects, each holding a_i objects initially
Output: The time required for moving the objects around so that all piles contain the same number of objects, subject to the constraint that only one pile moves one object to a neighbouring pile per unit time

Problem 11. TRANSPORT II
Input: N piles of objects, each holding a_i objects initially
Output: The time required for moving the objects around so that all piles contain the same number of objects, subject to the constraint that every pile moves at most one object to a neighbouring pile per unit time

2 Divide and Conquer

Problem 12. L-BLOCKS [1003]
Input: A marked cell (r, c) on a $2^N \times 2^N$ checker board
Output: A covering of the unmarked cells with L blocks

Problem 13. MODULAR EXPONENTIATION [20374]
Input: Integers A, B and M
Output: Value of $A^B \bmod M$

Problem 14. JOSEPHUS PROBLEM [1043]
Input: The number N of people sitting in a circle and an integer K
Output: The surviving position when every K people is killed