

LAB Assignment 5
CS205
Topic: Sorting and Selection
Lab Assignment : a Home Assignment: b

- a. The government has built a new city with n houses. These houses need to be furnished before they are allocated to people. A contractor is appointed to take care of furnishing all the n houses. The contractor knows the profit P_i that can be obtained from furnishing the i -th house. Assume that each P_i is a positive integer in the range $1 \dots M$, where M is the maximum profit that can be achieved from a single house. Contractor wants to allocate the duty to his two sons in such a way that his first son bags a profit S_1 , and his second son bags a profit S_2 . Let $T = P_0 + P_1 + \dots + P_{n-1}$ be the total profit from all the houses. So $T = S_1 + S_2$. Mr. Contractor loves his first son twice as much as his second son. Therefore, he likes to ensure that S_1 is as close to $2S_2$ as possible. Ideally, we should have $S_1 = 2T / 3$ and $S_2 = T / 3$, but this is not necessarily achievable, since there need not exist a subset of houses, the total profit from which is exactly $2T / 3$. So Contractor plans to divide the houses to his two sons in such a way that the absolute difference $|S_1 - 2S_2|$ is as small as possible.

Since there are 2^n subsets of houses, computing the sums of profits for all these subsets leads to an exponential amount of work. Contractor uses a dynamic-programming algorithm instead. He builds a two-dimensional table $A[i, j]$, where i runs from 0 through $n-1$, and j from 0 through T . The element $A[i, j]$ stores the possibility whether a profit of j (exactly) can be achieved from a subset of Houses 0, 1, 2,..., i . He initializes the first row $A[0, j]$ suitably. For $i \geq 1$, he computes $A[i, j]$ as follows. If $A[i-1, j]$ stores a non-zero value, then the profit j can be achieved from Houses 0, 1, 2,..., $i-1$. If so, he stores a non-zero value in $A[i, j]$. Otherwise, if $A[i-1, j - P_i]$ is a non-zero value, then by including House i the exact profit j can be achieved. In this case too, he stores a non-zero value at $A[i, j]$. If both the above conditions fail, there is no way to achieve the exact profit j from Houses 0, 1, 2,..., i and so he sets $A[i, j]$ to 0.

After the entire array A is prepared, Contractor looks at the last row to obtain the best partitioning of the houses to his two sons.

- I. Write a program to implement Contractor's algorithm.
- II. Read the number of houses (n) and the maximum profit bound for one house (M) from the user.
- III. Populate the profit array $P []$ with random integer entries in the range $1, 2, \dots, M$.
- IV. Initialize the d-p array A (its first row).
- V. Use Contractor's algorithm to populate the remaining rows of A . For your convenience, you may use two separate non-zero values to differentiate between the two cases that make an entry in A non-zero.
- VI. From the last row of A , decide the sums S_1 and S_2 which minimize the imbalance $|S_1 - 2S_2|$.

VII. Compute a subset (this need not be unique) of houses for the first son so that his total profit from this subset is exactly S_1 . The remaining houses go to the second son. Print the list of houses for the two sons.

Sample Output:

Enter number of houses (n) : 20

Enter profit bound (M) : 1000000

Profits are 618470 562398 186275 346181 972780 668101 188523 288997 866492
711633 80067 928084 299838 852255 232332 280079 180655 525332 707770 801006

Profit for first son : 6864839

Profit for second son : 3432429

Imbalance ($S_1 - 2 \cdot S_2$) : -19

First son gets houses 0 2 3 4 7 8 9 11 12 14 16 17 18

Second son gets houses 1 5 6 10 13 15 19

Upload Ass5a.c

- b) A pathologist was matching DNAs using its components such as Adenine, Cytosine, etc. He identified that there are total 26 components represented by 26 letters of the English alphabet. Given a sequence of Gene components, his task was to return the similarity between the two components. He considered the following similarity metric

Similarity: $\text{sum}(\text{len}(G_1), \text{len}(G_2)) - \text{Dis}(G_1, G_2)$, where Dis is the minimum modification distance between two gene sequences. Following three modification operations are allowed - insert, replace and delete and each operation costs 1. Modification distance between two strings is defined as minimum total cost incurred to convert one string to other.

Effect of three operations can be understood from the following example. Let us say the initial string (gene) is ABCDD:

Insert F at the end == ABCDDF

Replace A with V == VBCDDF

Delete F from end == VBCDD

For given genes: ABCGT and GTBAC

Similarity = $10 - 5 = 5$ where 10 is the sum of length of both the string and 5 is the minimum number of required operations.

Given a pair of gene sequences calculate the Similarity using memoization.

INPUT FORMAT:

Enter gene1= ABCGT

Enter gene2= GTBAC

Output:

Similarity: 5

Total number of operations required = 5

Upload Ass5b.c