TIME ANALYSIS AND OBSERAVATIONS

01 KNAPSACK PROBLEM

Algorithm 1:

Given n items, and each item has a weight wi and profit pi where i ranges from 1 to n, we **generate all possible subsets** of the items and check where all the sum of weights is less than the capacity of knapsack, among all such subsets we pick the option having maximum profit.

|  |  |  |
| --- | --- | --- |
| N  (No. of items) | C  (Capacity of sack) | Algo 1  Time (s) |
| 5 | 10 | 0.001 |
| 5 | 100 | 0.001 |
| 5 | 1000 | 0.001 |
| 5 | 10000 | 0.001 |
| 10 | 10 | 0.001 |
| 10 | 1000 | 0.001 |
| 10 | 100000 | 0.001 |
| 15 | Any C | 0.007 |
| 16 | Any C | 0.015 |
| 17 | Any C | 0.02 |
| 18 | Any C | 0.04 |
| 19 | Any C | 0.07 |
| 20 | Any C | 0.14 |
| 21 | Any C | 0.27 |
| 22 | Any C | 0.54 |
| 23 | Any C | 1.1 |
| 24 | Any C | 2.2 |
| 25 | Any C | 5 |
| 26 | Any C | 10 |
| 27 | Any C | 20 |
| 28 | Any C | 40 |
| 29 | Any C | 80 |
| 30 | Any C | 180 |
| 35 | Any C | 400 |

Observations

1. Higher values of N take a long time for computation. (Upto months for N>50)
2. Time values are independent of C. Hence for a given value of N, irrespective of C, the time taken is same
3. O(2^N), exponential curve

Algorithm 2:

Since each item is either picked or not picked, for each item, a choice needs to be made:

We consider two cases here,

1. When item is not chosen
2. When item is chosen

We find maximum weight in both cases and whichever is higher,

accordingly pick or discard the item.

1. When item is not chosen – Max profit = Maximum profit obtained by n-1 items and C weight (excluding nth item).
2. When item is chosen – Max profit = Profit of nth item plus maximum profit obtained by n-1 items and C minus weight of the nth item (including nth item).

C here denotes the capacity of knapsack. We use dynamic programming for coding this solution.

|  |  |  |
| --- | --- | --- |
| N  (No. of items) | C  (Capacity of sack) | Algo2  Time (s) |
| 5 | 10 | 0.001 |
| 5 | 10000 | 0.005 |
| 5 | 1000000 | 0.035 |
| 5 | 10000000 | 0.25 |
| 10 | 10 | 0.001 |
| 10 | 10000 | 0.006 |
| 10 | 1000000 | 0.065 |
| 10 | 10000000 | 0.5 |
| 100 | 10 | 0.001 |
| 100 | 1000 | 0.006 |
| 100 | 10000 | 0.025 |
| 100 | 1000000 | 0.5 |
| 1000 | 10 | 0.003 |
| 1000 | 1000 | 0.025 |
| 1000 | 10000 | 0.07 |
| 1000 | 100000 | 0.5 |
| 10000 | 10 | 0.01 |
| 10000 | 100 | 0.025 |
| 10000 | 1000 | 0.075 |
| 10000 | 10000 | 0.55 |
| 100000 | 1 | 0.02 |
| 100000 | 10 | 0.03 |
| 100000 | 100 | 0.07 |

Observations

1. All values are calculated within 1s
2. The times depends on the product of N and C, hence time for N = 1000, C = 1000, is almost same as N = 10000, C = 100.
3. O(NC)
4. For a given capacity, if the individual weights of items are kept low, more items can be accommodated in the sack, and it is observed that time increases slightly.