

Practical No-1

Page No.	
Date	

Date

Problem Statement :-

Write a program to calculate fibonacci number & find its step count.

Objective :-

- ① fibonacci numbers are used throughout society. It is astonishing how these sets of never-ending numbers are used in various ways.
- ② fibonacci numbers are very unique compared to the other mathematical subjects.

Theory :-

Fibonacci Numbers

The fibonacci frequency is a set of integers that start with zero followed by an one by another one & then by a series of steadily ~~que~~ increasing numbers. Their sequence follows the rule that each number is equal to the sum of the preceding two numbers.

The fibonacci sequence begin with the following K integers 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, -----

Calculating the Fibonacci Sequence

The fibonacci sequence can be calculated mathematically. In this approach, each number in the sequence is considered a term, which is represented by the expression F_n .

The 'n' reflects the numbers position in the sequence, starting with zero. for example, the sixth term is referred

to as F_5 , & the seventh term is referred to as F_6 .

- $F_0 = 0$ (applies only to first integer)
- $F_1 = 1$ (applies only to second integer)
- $F_n = F_{n-1} + F_{n-2}$ (applies to all other integer)

Important Notes on Fibonacci Numbers.

- Here is a list of a few points should be remembered while studying the fibonacci numbers.
- The concept of fibonacci numbers is only applicable to whole numbers & decimal numbers from a financial expression.
- The first fibonacci number is always 0 & the second fibonacci number is always 1.

Conclusion :-

Thus, we have studied about the fibonacci numbers.

Practical No-2

Page No.	
Date	

Page

Problem Statement :-

Implement job sequencing with deadlines using greedy method.

Objective :-

To find the sequence of job, which is completed within their deadlines & gives maximum profit. If a set of 'n' jobs are given which are associated with deadlines & profit is earned & job is completed by its deadline.

Theory :-

Given an array of jobs where every job has a deadline & associated profit, if the job is finished before the deadline. It is algorithm given that every job takes single unit time, so the minimum possible deadline for any job is 1. How to maximize total profit if only one job can be scheduled at a time.

Example :-

Input : four jobs with following deadline & profits

JobId	Deadlines	Profit
a	4	20
b	1	10
c	1	40
d	1	30

Output : c, a.

Greedy Approach for job sequencing problem :-

Greedy choose the jobs with maximum profit first, by sorting the jobs in decreasing order of their profit. This would help to maximize the total profit as choosing the job with maximum profit for every time slot ~~at~~ will eventually maximum the total profit.

Conclusion :-

Thus, we implemented job sequencing with deadlines using a greedy method which is completed within their deadlines & gives maximum profit.

Practical No - 3

Page No.	
Date	

Page

Problem Statement :-

Write a program to solve a fractional knapsack problem using a greedy method.

Objective :-

A greedy algorithm makes greedy choices at each step to ensure that the objective function is optimized.

To fill the knapsack of some given ~~va~~ volume with the different materials such that the value of selected items is maximized.

Theory :-

The Greedy algorithm are simple & straight forward. They are short sighted in their approach in the sense that they take decisions on the information at hand without worrying about the effect these decisions may have in the future they are easy to invent, easy to implement & most of the time quite efficient. Many problems cannot algorithms are used to solve optimization problems.

Greedy Approach :- Greedy algorithms works by making the decisions that seems most promising at any moment. It never reconsiders this decisions, whatever situation may arise later.

Knapsack Problem :-

There are two versions of problems.

① Fractional knapsack Problem

It is also one of the technique which are used to solve the knapsack problem. In fractional knapsack problem, the items are broken in order to maximize the profit. The problem in which we break the item is known as fractional knapsack problem.

② 0-1 knapsack problem.

The 0/1 knapsack problem means that the items are either completely or no item are having weight 2kg & 3kg respectively. If we pick the 2 kg item then we cannot pick 1 kg item from 2 kg item. We have to pick the 2kg item completely.

Conclusion :-

Thus, we studied about knapsack problem using a greedy method.

Practical No-4

Problem Statement :-

Write a program to generate a binomial coefficient using dynamic programming.

Objective :-

The binomial coefficient is used to denote the number of possible ways to choose a subset of objects of a given numerosity from a larger set.

It is so called because it be used to write the coefficients of the expansion of a power of a binomial.

Theory :-

Binomial coefficient is the binomial theorem which is an arithmetic expression. It is denoted as (N, K) which is equal to $N! / (K! * (N-K)!)$ where $!$ denotes the factorial.

This follows a recursive relation using we will calculate the N binomial coefficient in linear time $O(N * K)$ using DP.

Binomial theorem is also called as binomial expansion ~~dot~~ do linear the powers in algebraic equations. Binomial theorem helps us to find the expanded polynomial will always contain one ~~or~~ more than the power you are expanding.

$$(x-a)^n = \sum_{k=0}^n (k^n) x^k a^{n-k}$$

where,

Σ = known as 'sigma notation' used for sum all the terms in expansion from $k=0$ to $k=n$.
 n = Positive integer power of algebraic equation.
 (k^n) = read as 'n choose k'

Binomial coefficient is calculate by computing according to following expansion

$$C(n, k) = \frac{n!}{(n-k)! * k!}$$

Conclusion :-

Thus, we have studied about generating binomial coefficient using dynamic programming.