Greedy and Dynamic programming

Tanjima Akhanda Mim
191-15-2455

Coin change(DP).

1.

```
#include <stdio.h>
#include <string.h>
int min(int a, int b){
	if(a > b){
		return b;
	}else{
		return a;
	}
}
int main(  ){
	int change, k = 4;
	int coin[ ] = {1, 2, 8, 10};
	scanf("%d", &change);
	int dp[change + 1];
	memset(dp, 0, sizeof(dp));
	for(int j = 1; j <= change; j++){
		int mini = (int) 1e7;
		for(int i = 0; i < k; i++)
			if(j >= coin[i])
				mini = min(mini,1 + dp[j - coin[i]]);
		dp[j] = mini;
	}
	printf("%d\n", dp[change]);
	return 0;
}
```

2.

```
#include <stdio.h>
#include <string.h>
int min(int a, int b){
	if(a > b){
		return b;
	}else{
		return a;
	}
}
int main(){
```

```
	int change, n = 3, coin [ ] = {1,5,10};
	scanf("%d", &change);
	int dp[change + 1];
	memset(dp, 0, sizeof(dp));
	for(int j = 1; j <= change; j++){
		int mini = (int) 1e7;
		for(int i = 0; i < n; i++)
			if(j >= coin[i])
				mini = min(mini, 1 + dp[j - coin[i]]);
		dp[j] = mini;
	}
	printf("%d\n", dp[change]);
	return 0;
}
```

0 1 knapsack:

1.

```
#include <stdio.h>
#include <string.h>
int max(int a, int b) {
	if(a > b){
		return a;
	}else{
		return b;
	}
}
int main(  ){
	int n, i, w, W;
	printf("Total Items: ");
	scanf("%d", &n);
	int wt[n], val[n];
	printf("Weight: ");
	for(int i = 0; i < n; i++)
		scanf("%d", &wt[i]);
	printf("value: ");
	for(int i = 0; i < n; i++)
		scanf("%d", &val[i]);
	printf("Knapsack Weight: ");
	scanf("%d", &W);
```

```
	int Knapsack[n + 1][W + 1];
	for (i = 0; i <= n; i++) {
		for (w = 0; w <= W; w++) {
			if (i == 0 || w == 0)
				Knapsack[i][w] = 0;
			else if (wt[i - 1] <= w)
				Knapsack[i][w] = max(val[i - 1] + Knapsack[i - 1][w - wt[i - 1]],
Knapsack[i - 1][w]);
			else
				Knapsack[i][w] = Knapsack[i - 1][w];
		}
	}
	printf("%d", Knapsack[n] [W]);
	return 0;
}
```

2

```
#include <stdio.h>
#include <string.h>
int max(int a, int b) {
	if(a > b)
		return a;
	else
		return b;
}
int main(  ){
	int n, i, w, W = 50;
	printf("Total Items: ");
	scanf("%d", &n);
	int wt[n], val[n];
	printf("Weight: ");
	for(int i = 0; i < n; i++)
		scanf("%d", &wt[i]);
	printf("value: ");
	for(int i = 0; i < n; i++)
		scanf("%d", &val[i]);
	int Knapsack[n + 1][W + 1];
	for (i = 0; i <= n; i++) {
		for (w = 0; w <= W; w++) {
			if (i == 0 || w == 0)
				Knapsack[i][w] = 0;
```

```
                else if (wt[i - 1] <= w)
                        Knapsack[i][w] = max(val[i - 1] + Knapsack[i - 1][w - wt[i - 1]],
Knapsack[i - 1][w]);
                else
                        Knapsack[i][w] = Knapsack[i - 1][w];
                }
        }
        printf("%d", Knapsack[n] [W]);
        return 0;
}
```

Coin Changing(Greedy Approach)

1.

```
#include<stdio.h>
int main( ){
        int change, coin[ ] = {1,7,7,10};
        printf("Change Amount: ");
        scanf("%d", &change);
        printf("Coin Need: \n");
        for(int i = 3; i >= 0; i--)
                if(change >= coin[i]){
                        int cnt = 0;
                        while(change >= coin[i]){
                                ++cnt;
                                change -= coin[i];
                        }
                        printf("%d coin %d times\n", coin[i], cnt);
        }
        return 0;
}
```

2.

```
#include<stdio.h>
void swap(int *p, int *q){
	int temp = *p;
	*p = *q;
	*q = temp;
}
int main( ){
	int n, change;
	printf("Enter Number of Coin: ");
	scanf("%d", &n);
	int coin[n];
	for(int i = 0; i < n; i++)
		scanf("%d", & coin[i]);
	for(int i = 0; i < n-1; i++)
		for(int j = 0; j < n-i-1; j++)
			if(coin[j] > coin[j+1])
				swap(&coin[j], &coin[j+1]);
	printf("Change Amount: ");
	scanf("%d", &change);
	printf("Coin Need: \n");
	for(int i = n - 1; i >= 0; i--)
		if(change >= coin[i]){
			int cnt = 0;
			while(change >= coin[i]){
				++cnt;
				change -= coin[i];
			}
			printf("%d coin %d times\n", coin[i], cnt);
		}

	return 0;
}
```

Fractional knapsack

```
#include <stdio.h>
#include <string.h>
struct Node{
	int weight;
```

```
	int profit;
	double ratio;
};
int min(int a, int b){
	if(a > b){
		return b;
	}else{
		return a;
	}
}
int main(){
	int n, w;
	printf("Total Items: ");
	scanf("%d", &n);
	struct Node product[n];
	printf("Weight [] = ");
	for(int i = 0; i < n; i++){
		scanf("%d", &product[i].weight);
	}
	printf("Value [] = ");
	for(int i = 0; i < n; i++){
		scanf("%d", &product[i].profit);
		product[i].ratio = (product[i].profit * 1.0) / product[i].weight;
	}
	for(int i = 0; i < n - 1; i++)
		for(int j = 0; j < n - i - 1; j++)
			if(product[j].ratio < product[j + 1].ratio){
				struct Node temp;
				temp.weight = product[j].weight;
				temp.profit = product[j].profit;
				temp.ratio  = product[j].ratio;
				product[j].weight = product[j + 1].weight;
				product[j].profit = product[j + 1].profit;
				product[j].ratio  = product[j + 1].ratio;
				product[j + 1].weight = temp.weight;
				product[j + 1].profit = temp.profit;
				product[j + 1].ratio  = temp.ratio;
			}
	printf("Knapsack Weight: ");
	scanf("%d",&w);
	int total_profit = 0;
	for(int i = 0; i < n; i++){
		int k = min(w, product[i].weight);
```

```
            w -= k;
            total_profit += k * product[i].ratio;
        }
        printf("Maximum Profit: %d tk\n", total_profit);
        return 0;
}
```

Fibonacci:

1.

```
#include<stdio.h>
int main(){
    int t, n;
    printf("Enter Any Number: ");
    scanf("%d", &n);
    int fibonacci[n + 1];
    fibonacci[0] = 0;
    fibonacci[1] = 1;
    for(int i = 2; i <= n; i++){
        fibonacci[i] = fibonacci[i - 1] + fibonacci[i - 2];
    }
    printf("Fibonacci Number: %d", fibonacci[n]);
    return 0;
}
```

2.

```
#include<stdio.h>
int main(){
    int t, n;
    printf("Test Case: ");
    scanf("%d", &t);
    for(int k = 1; k <= t; k++){
        printf("Number %d: ", k);
        scanf("%d", &n);
        int fibonacci[n + 1];
        fibonacci[0] = 0;
        fibonacci[1] = 1;
        for(int i = 2; i <= n; i++){
            fibonacci[i] = fibonacci[i - 1] + fibonacci[i - 2];
        }
        printf("Fibonacci: %d\n", fibonacci[n]);
```

```
    }
    return 0;
}
```