|  |  |
| --- | --- |
| **EX.NO.:8** | **SEARCH THE KTH SMALLEST ELEMENT USING SELECTION SORT** |
| **DATE : 31.01.2025** |

**AIM:**

To search the Kth Smallest element in an Unsorted Array using Selection Sort

**ALGORITHM:**

STEP 1: Start the Program

STEP 2: Write a function kthsmallest(arr[],n,k) to search for the element in a position k within the array arr with the maximum size n.

STEP 3: Compare every element with all other elements in the array

STEP 4: Sort the array arr till k elements are placed correctly and return the index.

STEP 5: Display the kth smallest element value and its position in the array arr.

STEP 6: Stop the execution

**PROGRAM CODING:**

#include <stdio.h>

// Function to find the Kth smallest element using Selection Sort

int kthsmallest(int arr [], int n, int k)

{

int i,minindex,t;

for (i = 0; i < k; i++) { // Only sort till K elements are placed correctly

minindex = i;

for (int j = i + 1; j < n; j++) {

if (arr[j] < arr[minindex]) {

minindex = j;

}

}

// Swap the found minimum element with the first element

t = arr[i];

arr[i] = arr[minindex];

arr[minindex] = t;

}

return arr[k - 1]; // Kth smallest element

}

int main( )

{

int arr[ ] = {7, 10, 44, 133, 20, 15};

int n = sizeof(arr) / sizeof(arr[0]);

int k = 3;

printf("The %dth smallest element is %d\n", k, kthsmallest(arr, n, k));

return 0;

}

**RESULT:**

Thus the above program executed successfully and the results are verified.

|  |  |
| --- | --- |
| **EX.NO.:9** | **Optimal solution for the given Knapsack Problem using Greedy Method** |
| **DATE : 31.01.2025** |

**AIM:**

To find the Optimal Solution for the given knapsack problem using Greedy Method

**ALGORITHM:**

STEP 1: Start the Program

STEP 2: Get the inputs of all items with their weights and profits

STEP 3: Calculate Pi/Wi of all items and sort the items in descending order of Pi/Wi

STEP 4: Based on the knapsack weight, add the fractional part of the item if required.

STEP 5: Display the knapsack maximum profit calculated value.

STEP 6: Stop the execution

**PROGRAM CODING:**

#include<stdio.h>

int main()

{

float weight[50],profit[50],ratio[50],Totalvalue,temp,capacity,amount;

int n,i,j;

printf("Enter the number of items :");

scanf("%d",&n);

for (i = 0; i < n; i++)

{

printf("Enter Weight and Profit for item[%d] :\n",i);

scanf("%f %f", &weight[i], &profit[i]);

}

printf("Enter the capacity of knapsack :\n");

scanf("%f",&capacity);

for(i=0;i<n;i++)

ratio[i]=profit[i]/weight[i];

for (i = 0; i < n; i++)

for (j = i + 1; j < n; j++)

if (ratio[i] < ratio[j])

{

temp = ratio[j];

ratio[j] = ratio[i];

ratio[i] = temp;

temp = weight[j];

weight[j] = weight[i];

weight[i] = temp;

temp = profit[j];

profit[j] = profit[i];

profit[i] = temp;

}

printf("Knapsack problems using Greedy Algorithm:\n");

for (i = 0; i < n; i++)

{

if (weight[i] > capacity)

break;

else

{

Totalvalue = Totalvalue + profit[i];

capacity = capacity - weight[i];

}

}

if (i < n)

Totalvalue = Totalvalue + (ratio[i]\*capacity);

printf("\nThe maximum value is :%f\n",Totalvalue);

return 0;

}

**RESULT:**

Thus the above program executed successfully and maximum profit value can be displayed.