**DAA Practical-4**

**Name:-Mo Atif Azmi**

**Batch:-A4\_B4\_51**

**Practical 4:** Implement maximum sum of subarray for the given scenario of resource allocation using the divide and conquer approach.

**Editable Code:**

#include <stdio.h>

#include <stdlib.h>

typedef struct {

int start;

int end;

int sum;

} Subarray;

int max(int a, int b) {

return (a > b) ? a : b;

}

Subarray maxCrossingSubarray(int arr[], int low, int mid, int high, int constraint) {

int left\_sum = 0, sum = 0, best\_left = -1;

for (int i = mid; i >= low; i--) {

sum += arr[i];

if (sum <= constraint && sum > left\_sum) {

left\_sum = sum;

best\_left = i;

**DAA Practical-4**

}

}

int right\_sum = 0;

sum = 0;

int best\_right = -1;

for (int j = mid + 1; j <= high; j++) {

sum += arr[j];

if (sum + left\_sum <= constraint && sum > right\_sum) { right\_sum = sum;

best\_right = j;

}

}

Subarray result;

if (best\_left == -1 && best\_right == -1) {

result.start = -1; result.end = -1; result.sum = 0; } else {

result.start = (best\_left == -1) ? mid + 1 : best\_left; result.end = (best\_right == -1) ? mid : best\_right; result.sum = left\_sum + right\_sum;

}

return result;

}

**DAA Practical-4**

Subarray maxSubarrayDC(int arr[], int low, int high, int constraint) { if (low > high) {

Subarray empty = {-1, -1, 0};

return empty;

}

if (low == high) {

if (arr[low] <= constraint) {

Subarray single = {low, low, arr[low]};

return single;

} else {

Subarray empty = {-1, -1, 0};

return empty;

}

}

int mid = (low + high) / 2;

Subarray left = maxSubarrayDC(arr, low, mid, constraint); Subarray right = maxSubarrayDC(arr, mid + 1, high, constraint); Subarray cross = maxCrossingSubarray(arr, low, mid, high, constraint);

Subarray best = left;

if (right.sum > best.sum) best = right;

if (cross.sum > best.sum) best = cross;

**DAA Practical-4**

return best;

}

int main() {

int n, constraint;

printf("Enter number of tasks: ");

scanf("%d", &n);

int \*resources = (int\*)malloc(n \* sizeof(int));

printf("Enter resources array:");

for (int i = 0; i < n; i++) scanf("%d", &resources[i]);

printf("Enter resource constraint: ");

scanf("%d", &constraint);

Subarray ans = maxSubarrayDC(resources, 0, n - 1, constraint);

if (ans.start == -1) {

printf("No feasible subarray found under the constraint.\n"); } else {

printf("Best subarray (indices %d to %d): sum = %d\n", ans.start, ans.end, ans.sum);

printf("Elements: ");

for (int i = ans.start; i <= ans.end; i++) {

printf("%d ", resources[i]);

}

**DAA Practical-4**

printf("\n");

}

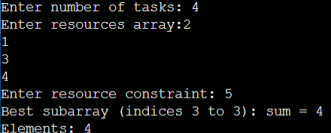
free(resources);

return 0;

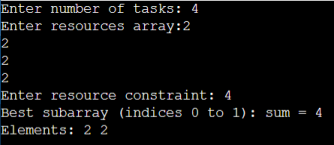
}

Output:

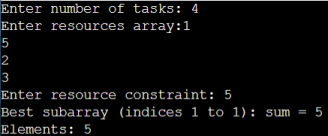
Case 1:



Case 2:

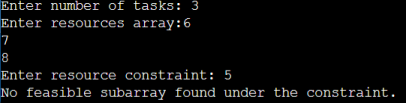


Case 3:

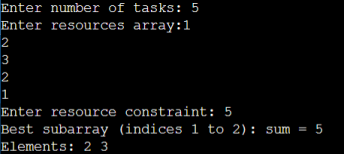


**DAA Practical-4**

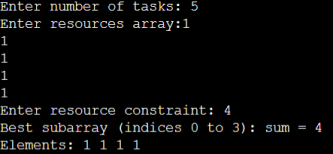
Case 4:



Case 5:

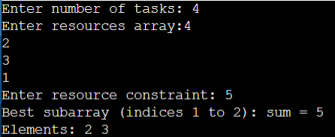


Case 6:



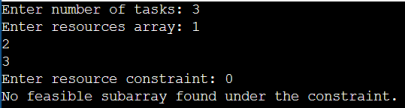
Case 7:

**DAA Practical-4**

****

Case 8:

Case 9:



Case 10:

