1. Implement fractional knapsack problem using Greedy technique.

#include <stdio.h>

#include <stdlib.h>

typedef struct {

int weight, value;

} Item;

int compare(const void \*a, const void \*b) {

double r1 = ((Item \*)b)->value / (double)((Item \*)b)->weight;

double r2 = ((Item \*)a)->value / (double)((Item \*)a)->weight;

return r1 > r2 ? 1 : -1;

}

double fractionalKnapsack(int W, Item arr[], int n) {

qsort(arr, n, sizeof(Item), compare);

double totalValue = 0.0;

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

if (arr[i].weight <= W) {

W -= arr[i].weight;

totalValue += arr[i].value;

} else {

totalValue += arr[i].value \* ((double)W / arr[i].weight);

break;

}

}

return totalValue;

}

int main() {

Item arr[] = {{10, 60}, {20, 100}, {30, 120}};

int W = 50;

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

printf("Maximum value in Fractional Knapsack = %.2f\n", fractionalKnapsack(W, arr, n));

return 0;

}

Output:

A black screen with white text

AI-generated content may be incorrect.