Binary Knapsack - Hackerearth

Monday, February 3, 2020 11:05 PM

The Code:

```
#include <stdio.h>
#include <math.h>


int max(int x, int y){
  return (x > y) ? x : y;
}

int binaryKnapsack(int length, int weight, int* values, int* weights, int** memo){

  int  valuesUsed, weightFilled, currentMemoIndex, previousMemoIndex;

  for(valuesUsed=1; valuesUsed<=length; valuesUsed++){

    currentMemoIndex = valuesUsed % 2;
    previousMemoIndex = 1 - currentMemoIndex;
    for(weightFilled=1; weightFilled<=weight; weightFilled++){

      if(weights[valuesUsed] <= weightFilled)
        memo[currentMemoIndex][weightFilled] = max(values[valuesUsed] +
memo[previousMemoIndex][weightFilled - weights[valuesUsed]],
                            memo[previousMemoIndex][weightFilled] );
      else
        memo[currentMemoIndex][weightFilled] = memo[previousMemoIndex][weightFilled];
    }
  }

  return memo[currentMemoIndex][weight];
}


int main(){
     int length, weight, iter, bestValue;
     scanf("%d %d", &length, &weight);

     int* values = (int*)calloc(length+1, sizeof(int));
     int* weights = (int*)calloc(length+1, sizeof(int));
     int** memo = (int**)calloc(2, sizeof(int*));

     for(iter=1; iter<=length; iter++) {
       scanf("%d", &values[iter]);
       if(iter <= 2) {
         memo[iter-1] = (int*)calloc(weight+1, sizeof(int));
       }
     }
```

```
        for(iter=1; iter<=length; iter++){
            scanf("%d", &weights[iter]);
        }

        bestValue = binaryKnapsack(length, weight, values, weights, memo);

        printf("%d", bestValue);

}
```