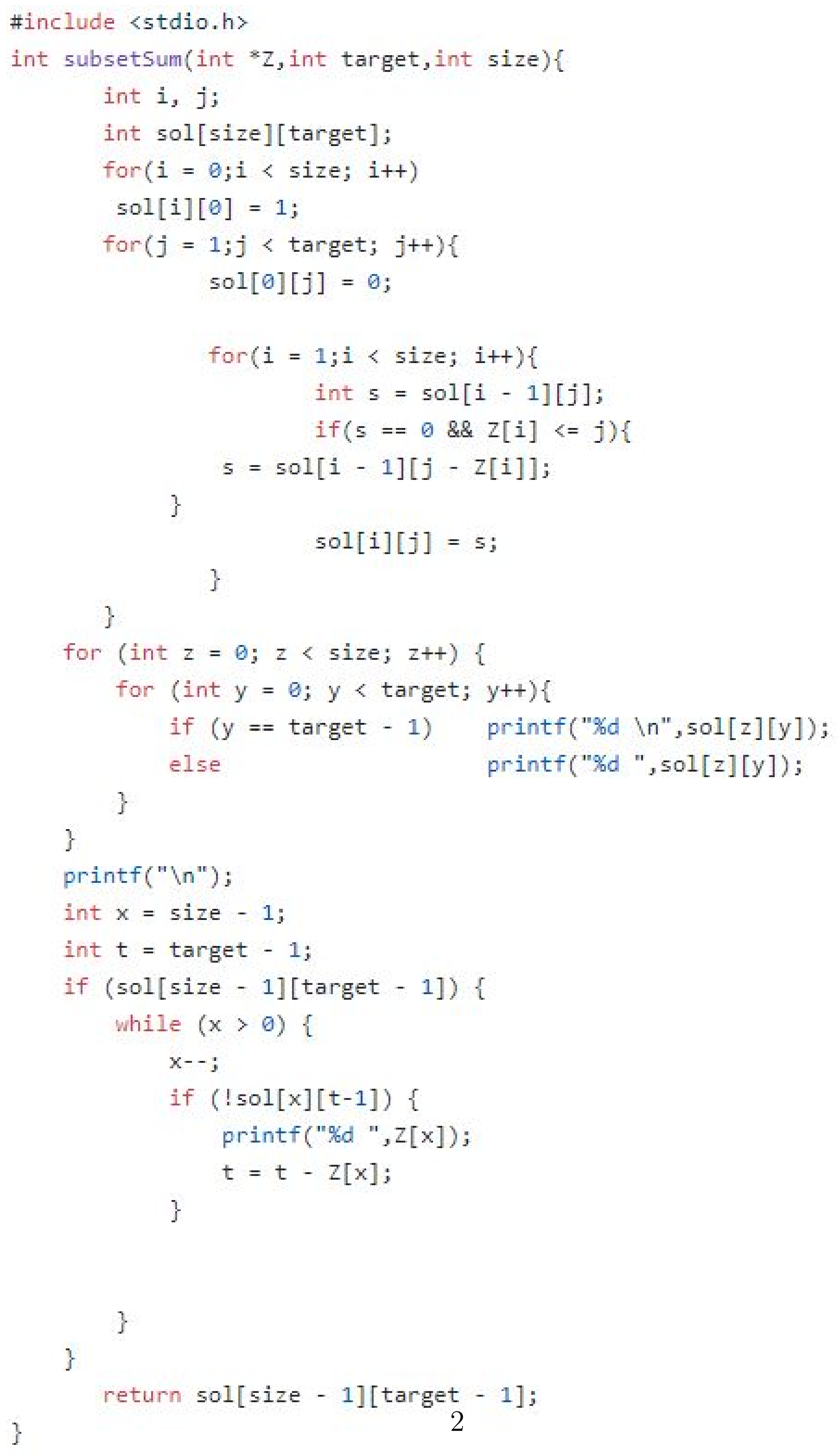
TCSS 343 - Challenge 3 - Subset Sum Problem

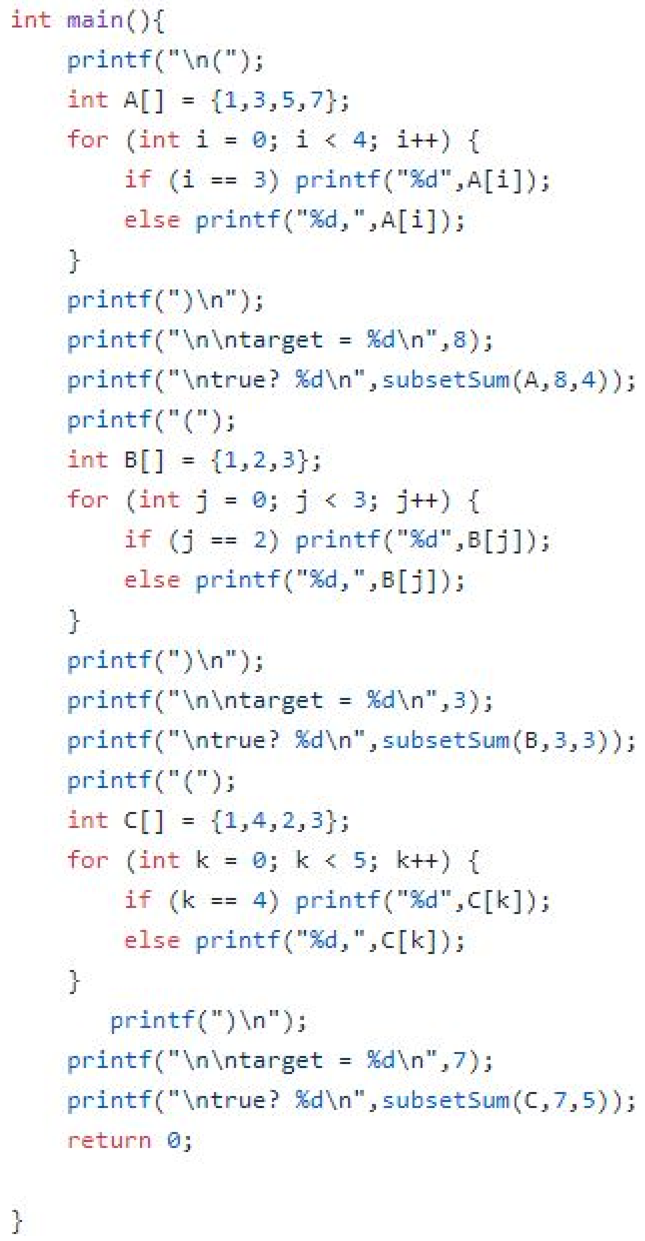
Jake McKenzie

August 8, 2018

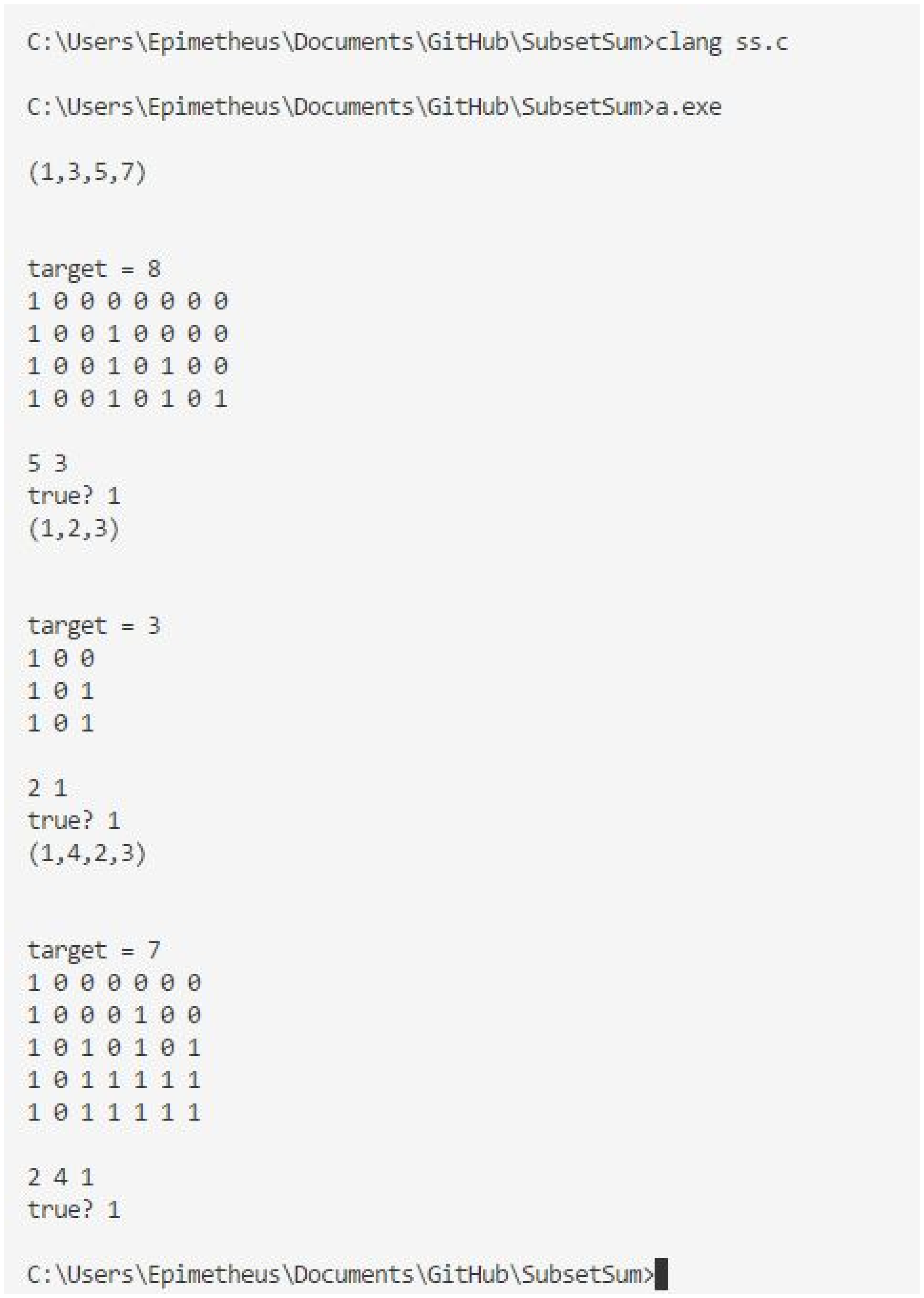
The key to dynamic programming, at least to me, is to truly write down what steps I need to solve a problem iteratively, ignore the typical recursive step and memoize. Eisenhower put it perfectly: “No battle was ever won according to plan, but no battle was ever won without one.” The first thing to notice is that an optimal dynamic program will have the property no matter the initial state and initial decisions, the corresponding choices the algorithm makes must be the optimal choice for each successive step forward. This is Bellman’s principle of optimality.



```
#include <stdio.h>
int subsetSum(int *Z,int target,int size){
       int i, j;
       int sol[size][target];
       for(i = 0;i < size; i++)
        sol[i][0] = 1;
       for(j = 1;j < target; j++){
               sol[0][j] = 0;

               for(i = 1;i < size; i++){
                       int s = sol[i - 1][j];
                       if(s == 0 && Z[i] <= j){
                s = sol[i - 1][j - Z[i]];
            }
                       sol[i][j] = s;
               }
       }
    for (int z = 0; z < size; z++) {
        for (int y = 0; y < target; y++){
            if (y == target - 1)    printf("%d \n",sol[z][y]);
            else                    printf("%d ",sol[z][y]);
        }
    }
    printf("\n");
    int x = size - 1;
    int t = target - 1;
    if (sol[size - 1][target - 1]) {
        while (x > 0) {
            x--;
            if (!sol[x][t-1]) {
                printf("%d ",Z[x]);
                t = t - Z[x];
            }


        }
    }
       return sol[size - 1][target - 1];
}
```



```c
int main(){
    printf("\n(");
    int A[] = {1,3,5,7};
    for (int i = 0; i < 4; i++) {
        if (i == 3) printf("%d",A[i]);
        else printf("%d,",A[i]);
    }
    printf(")\n");
    printf("\n\ntarget = %d\n",8);
    printf("\ntrue? %d\n",subsetSum(A,8,4));
    printf("(");
    int B[] = {1,2,3};
    for (int j = 0; j < 3; j++) {
        if (j == 2) printf("%d",B[j]);
        else printf("%d,",B[j]);
    }
    printf(")\n");
    printf("\n\ntarget = %d\n",3);
    printf("\ntrue? %d\n",subsetSum(B,3,3));
    printf("(");
    int C[] = {1,4,2,3};
    for (int k = 0; k < 5; k++) {
        if (k == 4) printf("%d",C[k]);
        else printf("%d,",C[k]);
    }
       printf(")\n");
    printf("\n\ntarget = %d\n",7);
    printf("\ntrue? %d\n",subsetSum(C,7,5));
    return 0;

}
```



```
C:\Users\Epimetheus\Documents\GitHub\SubsetSum>clang ss.c

C:\Users\Epimetheus\Documents\GitHub\SubsetSum>a.exe

(1,3,5,7)


target = 8
1 0 0 0 0 0 0 0
1 0 0 1 0 0 0 0
1 0 0 1 0 1 0 0
1 0 0 1 0 1 0 1

5 3
true? 1
(1,2,3)


target = 3
1 0 0
1 0 1
1 0 1

2 1
true? 1
(1,4,2,3)


target = 7
1 0 0 0 0 0 0
1 0 0 0 1 0 0
1 0 1 0 1 0 1
1 0 1 1 1 1 1
1 0 1 1 1 1 1

2 4 1
true? 1

C:\Users\Epimetheus\Documents\GitHub\SubsetSum>
```