

Assignment-6

Dynamic Programming: Subset Sum

Subset Sum

- You are given an array $A = (a_1, a_2, a_3, \dots, a_n)$ of n positive integers. If you write the integers in the order they appear in A , and insert symbols $+$ or $-$ before each integer, you get an arithmetic expression. If the evaluation of the expression gives the value v , we say that v is realized (or realizable) by the array A . For example, consider the array $(7, 12, 1, 9, 5)$ of five positive integers. We have
 - $+7-12-1+9+5 = 8$,
 - $-7-12+1-9+5 = -22$,
 - $-7+12-1-9+5 = 0$,
- that is, the integers $8, -22, 0$ are realizable by this array. You are given a target value T . Your task is to find out whether T is realizable by the given array A .

Subset Sum: Part 1

- Write a function **realizable(A, n, T)** to decide whether T is realizable by the array A of size n.
- The function should implement a dynamic-programming approach.
- Build a two-dimensional table $P[0 \dots n][-S \dots S]$ such that $P[i][j]$ would store the decision whether the value j can be realized by the prefix $(a_1, a_2, a_3, \dots, a_i)$ of A.
- For $i=0$, we consider no elements from A, so the only realizable value is 0.
- For $i>1$, the value j is realizable by $(a_1, a_2, a_3, \dots, a_i)$ if and only if either $j-a_i$ or $j+a_i$ is (or both are) realizable by $(a_1, a_2, a_3, \dots, a_{i-1})$.
- Use this recursive formulation to build the table P in $O(nS)$ time. The final decision is available as $P[n][T]$.

Subset Sum: Implementation Issues

- Each row of P should store $2S+1$ decisions, and is indexed in the range $[-S, S]$.
- In C/C++, negative indexing may lead to devastating consequences, and must be avoided.
- Elements of a row $P[i]$ of size $2S+1$ in a two-dimensional C/C++ array $P[][]$ are indexed in the range $[0, 2S]$. This means that the logical quantity $P[i][j]$ with j in $[-S, S]$ is to be found in the physical location $P[i][j+S]$.
- The next issue pertains to the table lookup at column indices $j \pm a_i$. If any of these indices is not in the range $[-S, S]$, the corresponding lookup should not be made.

Subset Sum: Part 2

- Copy the function of Part 1 to a function **showone(A, n, T)** that follows the same algorithm as Part 1 but additionally prints one way of realizing T (in case T is realizable) out of many ways. The running time of this function should be $O(nS)$.
- The `main()` function
- Read $n, a_1, a_2, a_3, \dots, a_n, T$ from the user.
- Call **realizable** to decide whether T is realizable by A.
- Call **showone** to print some realization of T (provided that T is realizable).
- Do not use memoization to solve Part 1 and Part 2.

Submission

- Last date: 4-OCT-2024 (till 11:59 P.M.) (Friday)
- Programming language: C/C++
- Single File: 24CS06001_A6.c/.cpp or 24AI06001_A6.c/.cpp
- Subject Line: 24CS06001_A6 or 24AI06001_A6
- Email to: pds2016autumn@gmail.com