ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

Solution to 2nd QUIZ SUMMER SEMESTER, 2019-2020

CSE 4403: Algorithms

Solve the following problem using dynamic programming. If the input needs to be preprocessed, mention it under the header **Preprocessing**. Then define a set of **Subproblems**, **Relate** the subproblems recursively, argue the relation is acyclic, provide **Base** cases, construct a **Solution** from the subproblems, and analyze running **Time**. Finally, if you need to perform some more calculations after getting the DP solution, mention it under the header **Postprocessing**.

1. (Adapted from CodeForces 455A)
One day two cats stole a bag of bread from the kitchen of a house. The bag contained $N(1 \leq N \leq 10^5)$ pieces of bread of various sizes. They agreed to divide the bread pieces into two equal parts but failed. The last time they faced this scenario, they went to a monkey, who, instead of dividing their bread pieces into equal halves, ate all of it. For this reason, they came to you. Taking the bag, you realized that the bag is magical. Whenever you pull out a bread piece of size X from the bag, all the bread pieces of size $(X-1)$ and size $(X+1)$ disappear. Now the cats want to know what is the maximum sum of the sizes you can make, by pulling out the bread pieces optimally. Propose a solution to the problem.
Solution:
Preprocessing

   - Calculate the frequency of each pieces of size i, $freq[i]$
   - Find out the maximum size of the bread pieces, mx

   Subproblem

   - $x(i) =$ Maximum sum of sizes that can be made by considering piece of size i

   Relate

   - Can either take the bread of size i: $freq[i] \times i+$Best from size $(i-2)$
   - Can leave the bread of size i: Best from size $(i-1)$
   - Take the best of these two
   - $x(i) = \max(freq[i] \times i + x(i-2), x(i-1))$
   - Only depends on smaller prefix, acyclic

   Base

   - Bread pieces of size 0 or less add 0 to the best sum
   - $x(0) = 0$, $x(-1) = 0$

   Solution

   - Find $x(mx)$

   Time

   - # of subproblems: mx
   - Work per subproblem: $O(1)$
   - Total Work: $O(mx)$

2. (Adapted from CodeForces 903A)
One day two cats stole a bag of bread from the kitchen of a house. The bag contained bread pieces of the following sizes: $7g$ and $3g$. They agreed to divide the bread pieces into two equal parts but failed. The last time they faced this scenario, they went to a monkey, who, instead of dividing their bread pieces into equal halves, ate all of it. For this reason, they came to you. Taking the bag, you realized that the bag is magical. It carries an infinite supply of bread pieces of the sizes specified above. Now the cats are interested to know, given any amount $N(1 \leq N \leq N)$

in grams, whether it's possible to make that amount using the bread pieces from the bag.
Propose a solution to the problem.
Solution:
Subproblems

- $x(i)$ = If it is possible to make amount i or not $\in [0, 1]$

Relate

- To make amount i, we need to check if it's possible to make amount $(i-3)$ or $(i-7)$
- If either one is possible, then i is possible
- $x(i) = OR(x(i-3), x(i-7))$
- Only depends on smaller prefix, acyclic

Base

- If $i < 0$, it's not possible to make that amount
- $x(<0) = False$
- It's possible to make amount 0 by not taking any bread
- $x(0) = True$

Solution

- Find $x(N)$

Time

- # of subproblems: N
- Work per subproblem: $O(1)$
- Total Work: $O(N)$

3. (Adapted from Gym 101982C)
One day two cats stole a bag of bread from the kitchen of a house. The bag contained $N(1 \leq N \leq 1000)$ pieces of bread of various sizes. They agreed to divide the bread pieces into two equal parts but failed. The last time they faced this scenario, they went to a monkey, who, instead of dividing their bread pieces into equal halves, ate all of it. For this reason, they came to you. Taking the bag, you realized that the bag is magical. So you want to keep $k(1 \leq k \leq n)$ bread pieces for yourself. But you want the sizes of each bread piece you take to be different. The cats agreed to your proposal, but in one condition. You need to figure out how many ways you can pick k bread pieces, each with different sizes. Changing the order of listing does not increase the count.
Solution:
Preprocessing

- Calculate the frequency of each pieces of size i, $freq[i]$
- Create a set of unique piece sizes, v
- Let, $|v| = cc$

Subproblems

- $x(i, j)$ = # of ways to take j pieces by considering bread pieces of size $v[i]$ and more

Relate

- Can either take the bread piece of size $v[i]$: $v[i]\times$ # of ways to take $j-1$ pieces by considering rest of the sizes
- Can leave the bread piece of size $v[i]$: # of ways to take j pieces by considering rest of the sizes
- $x(i, j) = v[i] \times x(i-1, j-1) + x(i-1, j)$
- Only depends on smaller prefix, acyclic

Base

- If we have already taken k pieces, then we won't take any more pieces
- $x(i, 0) = 0$
- If we run out of pieces, we can't take any more pieces
- $x(0, j) = 0$

Solution

- Find $x(cc - 1, k)$

Time

- # of subproblems: $O(cc \times k)$
- Work per subproblem: $O(1)$
- Total Work: $O(cc \times k)$

4. (Adapted from UVA 147)
One day two cats stole a bag of bread from the kitchen of a house. The bag contained bread pieces of the following sizes: $1000g$, $500g$, $100g$, $50g$, $20g$, $10g$, $5g$, $2g$, and $1g$. They agreed to divide the bread pieces into two equal parts but failed. The last time they faced this scenario, they went to a monkey, who, instead of dividing their bread pieces into equal halves, ate all of it. For this reason, they came to you. Taking the bag, you realized that the bag is magical. It carries an infinite supply of bread pieces of the sizes specified above. Now the cats are interested to know, given any amount $N (1 \leq N \leq 30)$ in kilograms, how many ways can that amount be made up by combining bread pieces of various sizes. Changing the order of listing does not increase the count.
Solution
Preprocessing

- Convert the total amount to gram by multiplying by 1000, k
- Let, val = The sizes of different bread pieces

Subproblems

- $x(i, s) =$ How many ways we can make the remaining amount s by considering the first i pieces

Relate

- If the remaining amount, s is greater than or equal to the current bread piece size, $val[i]$, we can take the piece, possibly multiple times: $x(i, s - val[i])$
- Also we can ignore the piece size, and move on to the next one: $x(i + 1, s)$
- $x(i, s) = x(i, s - val[i])$ if $s \geq val[i] + x(i + 1, s)$
- Only depends on smaller suffix, acyclic

Base

- If we run out of piece sizes, we're done
- $x(10, s) = 0$
- If the remaining size is 0, we're done
- $x(i, 0) = 0$

Solution

- Find $x(0, N)$

Time

- # of subproblems: $O(N)$
- Work per subproblem: $O(1)$
- Total Work: $O(N)$

5. (Adapted from UVA 562)
One day two cats stole a bag of bread from the kitchen of a house. The bag contained $N(1 \leq N \leq 100)$ pieces of bread of various sizes. They agreed to divide the bread pieces into two equal parts but failed. The last time they faced this scenario, they went to a monkey, who, instead of dividing their bread pieces into equal halves, ate all of it. For this reason, they came to you. Propose a solution to determine the fairest division between two cats. This means that the difference between the total sizes of the bread each cat obtains should be minimized. The size of the bread varies from $1g$ to $500g$. It's not allowed to split a single piece of bread. Your task is to find out the minimal positive difference between the total size of the bread the two cats obtain when you divide the bread pieces from the corresponding bag.
Solution:
Preprocessing

- Take the sum of all possible bread sizes, $total$
- Let, arr = The sizes of different bread pieces

Subproblems

- $x(i, R) =$ Maximum amount made by taking the first i pieces when the remaining amount is R

Relate

- If the remaining amount, R is greater than or equal to the size of the current bread piece size, $arr[i]$, we can take the piece: $arr[i] + f(i+1, R - arr[i])$
- If the remaining amount, R is less than the size of the current bread piece size, $arr[i]$, we can get: 0
- We can ignore the piece size, $arr[i]$: $f(i+1, R)$
- We'll take the best of the two
- $x(i, R) = \max(arr[i] + f(i+1, R - arr[i])$ if $R \geq arr[i]$, else $0, f(i+1, R))$
- Only depends on smaller suffix

Base

- If we run out of piece sizes, we're done
- $x(N+1, R) = 0$
- If the remaining size is 0, we're done
- $x(i, 0) = 0$

Solution

- Lab Assignment 2: For the fairest division, we can get $\left\lceil \frac{total}{2} \right\rceil$
- Find $x\left(0, \left\lceil \frac{total}{2} \right\rceil\right)$

Time

- # of subproblems: $O(total)$
- Work per subproblem: $O(1)$
- Total Work: $O(total)$

Postprocessing

- Lab Assignment 2: The difference will be: $total - 2 \times x\left(0, \left\lceil \frac{total}{2} \right\rceil\right)$