

Question 1

You have n identical flowers, which you would like to place into k vases, according to the following rules:

- Each flower must be placed in a vase.
- The i th vase (where $1 \leq i \leq k$) must hold at least 1 and at most f_i flowers.
- No two vases contain the same number of flowers.

Your goal is to assign a number of flowers to each vase following the above rules, or to determine that no such assignment exists.

1.1 [10 marks] Suppose that $f_1 = \dots = f_k = n$. Design an algorithm which runs in $O(k)$ time and achieves the goal.

Analysis:

Suppose k is the number of vases

According to the topic, in order to achieve the condition that all the vases must have at least one flowers and the number of flowers in each vase varies, the minimum sum number of flower is $1 + 2 + \dots + k$.

If $1 + 2 + \dots + k > n$, it means that at least two vase will have same number flowers or some vases is empty.

If $1 + 2 + \dots + k \leq n$, suppose p is the number of extra flowers at the end of the allocation when the allocation is done as described above, if there is nothing left, then $p = 0$. now it need to prove that both $1 + 2 + \dots + k + p = n (p \geq 0)$ and $f_k = n \geq (k + p)$ is true.

$$\begin{aligned}
 1 + 2 + 3 + \dots + k + p &= \frac{(1+k)k}{2} + p = n \geq (k + p) \\
 \frac{(1+k)k}{2} + p &\geq k + p \\
 (1+k)k &\geq 2k \\
 k^2 - k &\geq 0 \\
 k &\geq 1 \text{ OR } k \leq 0
 \end{aligned}$$

According to the above, it works as long as it's bigger than one vase. Therefore, when $1 + 2 + \dots + k \leq n$ is true.

Answer:

From the first vase, assign i flowers to the i th vase.

During the assignment, if the flowers are used up or the last vase assign less than k flowers, it means that the total flower cannot assign by the topic.

When the assignment finished, if none flower left, the method is true.

If there are any flowers left over, put the rest directly into the last bottle, according to the above proof, n must larger than the number of flowers in the last vase.

The method only traverses each vase once, so the time complexity is $O(k)$

1.2 [10 marks] Now we return to the original problem, in which the f_i can take any positive integer values. Design an algorithm which runs in $O(k \log k)$ time and achieves the goal.

Answer:

First, put all of the f_i in an array, Suppose the array is A .

$$A = [f_1, f_2, \dots, f_k]$$

Second, use Merge Sort to sort the array A , suppose the new array is A' , The merge sort's time complexity is $O(k \log k)$.

Third, Create an new array N which record the number of flower in each vase which focus on the sorted array A' . Suppose S_{min} is the least sum number of flower which had been assigned, k is the number of vases

Now, suppose the index i is the position of vase in sorted array A' , $N[i]$ record the number of flower which will assign in i th vase.

According to the topic, each vase must have at least one flower ($A'[1] \geq 1$), assign the number of flower in first vase is 1 ($N[1] = 1$), $S_{min} = 1$.

Perform the following step while $i > 1$ AND $i \leq k$:

- If $N[i - 1] + 1 \leq A'[i - 1]$, set $N[i] = N[i - 1] + 1$, $S_{min} = N[i] + S_{min}$, continue
- If $N[i - 1] + 1 > A'[i - 1]$, it means that existed at least two vases with same number of flower. this situation cannot assign by the topic.
- If $Sum_{min} > n$, it means that there is at least one vase cannot assign flower, this situation cannot assign by the topic.

This part's count the least number of flower in this f_i 's situation. The time complexity is $O(k)$

Suppose S_{max} is the most sum number of flower which had been assigned. According to A' , when $i = k$, the $A'[i]$ is the largest number of the maximum flower in the vase. The last vase is $N[i] = A'[i]$, $S_{max} = N[i]$

Perform the following step while $i < k$ AND $i \geq 1$:

- If $N[i + 1] - 1 \leq A'[i]$, set $N[i] = N[i + 1] - 1$, $S_{max} = S_{max} + N[i]$, continue
- If $N[i + 1] - 1 > A'[i]$, set $N[i] = A'[i]$, $S_{max} = S_{max} + N[i]$, continue
- If $N[i] < 1$, the situation is not follow the topic, this situation cannot assign.

This part's count the most number of flower in this f_i 's situation. The time complexity is $O(k)$

Compare the n with S_{min} and S_{max}

- If $S_{min} \leq n \leq S_{max}$, this f_i 's situation of vases can assign n flower.
- If $n < S_{min}$, the situation must exist at least one vases have same number with other flower or at least one vase cannot assign any flower.
- If $n > S_{max}$, this situation must exist at least one flower cannot assign in each vases or at least one vases assign the number of flower is larger than the setting f_1

The total time complexity is $O(k \log k)$