COMP108
Data Structures and Algorithms

Data structures - Arrays (Part III 2D Arrays)

Professor Prudence Wong

pwong@liverpool.ac.uk

2022-23

Maximum Monthly Average Rainfall over a Year

- Input: Suppose we have data of daily rainfall for a year.
- Output: Find the maximum monthly average rainfall over a year.

Maximum Monthly Average Rainfall over a Year

- Input: Suppose we have data of daily rainfall for a year.
- Output: Find the maximum monthly average rainfall over a year.
- Solutions:
  - For simplicity, let's assume there are d days in a month and m months in a year.

Maximum Monthly Average Rainfall over a Year

- Input: Suppose we have data of daily rainfall for a year.
- Output: Find the maximum monthly average rainfall over a year.
- Solutions:
  - For simplicity, let's assume there are d days in a month and m months in a year.
  - Store rainfall data in a 2D array of size $m \times d$.

	1	2	3	$\cdots$	d
1					
2					
3					
$\vdots$					
m					

  - *rainfall*$[i][j]$ stores rainfall of month i and day j

Maximum Monthly Average Rainfall over a Year

- Input: Suppose we have data of daily rainfall for a year.
- Output: Find the maximum monthly average rainfall over a year.
- Solutions:
  - For simplicity, let's assume there are d days in a month and m months in a year.
  - Store rainfall data in a 2D array of size $m \times d$.

	1	2	3	$\cdots$	d
1					
2					
3					
$\vdots$					
m					

```
1 <= i <= m
1 <= j <= d
```

  - *rainfall*$[i][j]$ stores rainfall of month i and day j
  - Sub-problem (i): what is average rainfall of month i?
  - Sub-problem (ii): what is the maximum average?

Sub-problem (i) Average Rainfall of month *i*

Sub-problem (i) Average Rainfall of month *i*

$j \leftarrow 1$

$sum \leftarrow 0$

$average \leftarrow sum / d$

Sub-problem (i) Average Rainfall of month *i*

$j \leftarrow 1$
$sum \leftarrow 0$
while $j \leq d$ do
begin

    $j \leftarrow j + 1$
end
$average \leftarrow sum/d$

Sub-problem (i) Average Rainfall of month *i*

$j \leftarrow 1$
$sum \leftarrow 0$
while $j \leq d$ do
begin
    $sum \leftarrow sum + rainfall[i][j]$
    $j \leftarrow j + 1$
end
$average \leftarrow sum/d$

Sub-problem (ii) Maximum of the averages

$j \leftarrow 1$
$sum \leftarrow 0$
while $j \leq d$ do
begin
    $sum \leftarrow sum + rainfall[i][j]$
    $j \leftarrow j + 1$
end
$average \leftarrow sum/d$

Sub-problem (ii) Maximum of the averages

$M \leftarrow 0$
$i \leftarrow 1$
while $i \leq m$ do
begin

$j \leftarrow 1$
$sum \leftarrow 0$
while $j \leq d$ do
begin
    $sum \leftarrow sum + rainfall[i][j]$
    $j \leftarrow j + 1$
end
$average \leftarrow sum/d$

    if $average > M$ then
        $M \leftarrow average$
    $i \leftarrow i + 1$
end
output M

Sub-problem (ii) Maximum of the averages

$M \leftarrow 0$
$i \leftarrow 1$
while $i \leq m$ do
begin
    $sum \leftarrow 0$
    for $j \leftarrow 1$ to d do
        $sum \leftarrow sum + rainfall[i][j]$
    $average \leftarrow sum/d$
    if $average > M$ then
        $M \leftarrow average$
    $i \leftarrow i + 1$
end
output M

$j \leftarrow 1$
$sum \leftarrow 0$
while $j \leq d$ do
begin
    $sum \leftarrow sum + rainfall[i][j]$
    $j \leftarrow j + 1$
end
$average \leftarrow sum/d$

Sub-problem (ii) Maximum of the averages (Time Complexity?)

```
M ← 0
i ← 1
while i ≤ m do
begin
    sum ← 0
    for j ← 1 to d do
        sum ← sum + rainfall[i][j]
    average ← sum/d
    if average > M then
        M ← average
    i ← i + 1
end
output M
```

```
j ← 1
sum ← 0
while j ≤ d do
begin
    sum ← sum + rainfall[i][j]
    j ← j + 1
end
average ← sum/d
```

Sub-problem (ii) Maximum of the averages (Time Complexity?) $O(md)$

```
M ← 0
i ← 1
while i ≤ m do
begin
    sum ← 0
    for j ← 1 to d do
        sum ← sum + rainfall[i][j]
    average ← sum/d
    if average > M then
        M ← average
    i ← i + 1
end
output M
```

```
j ← 1
sum ← 0
while j ≤ d do
begin
    sum ← sum + rainfall[i][j]
    j ← j + 1
end
average ← sum/d
```

Summary: 2D Arrays

Next: Stacks/Queues

For note taking