Recap

Bin packing

- Given bins of a fixed scalar size, and a set of objects, pack the objects into as few bins as possible.

- Solve problems like this in shipping, in using computer memory or disk space, etc.

Recap – heuristics

Consider items in given order

Put each item in the first bin where it fits

- Called: heuristic FF
Sample case

Bins are size 25.
Items are: \{13, 9, 8, 4, 7, 11, 7, 10, 6\}

How many bins does FF give?
A. 3
B. 4
C. 5

Sample case

Bins are size 25.
Items are: \{13, 9, 8, 4, 7, 11, 7, 10, 6\}

FF:
Bin 1: 13, 9
Bin 2: 8, 4, 7, 6
Bin 3: 11, 7
Bin 4: 10
Recap – heuristics
Consider items in sorted order, biggest first
Put each item in the first bin where it fits
  • Called: heuristic FFD

Sample case
Bins are size 25.
Items are: {13, 9, 8, 4, 7, 11, 7, 10, 6}

How many bins does FFD give?
A. 3
B. 4
C. 5
Sample case

Bins are size 25.
Items are: {13, 9, 8, 4, 7, 11, 7, 10, 6}

FFD:
Bin 1: 13, 11
Bin 2: 10, 9, 6
Bin 3: 8, 7, 7
Bin 4: 4

Sample case

Bins are size 25.
Items are: {13, 9, 8, 4, 7, 11, 7, 10, 6}

How many bins is optimal?
A. 3
B. 4
C. 5
Sample case

Bins are size 25.
Items are: {13, 9, 8, 4, 7, 11, 7, 10, 6}

Optimal:
Bin 1: 13, 8, 4
Bin 2: 9, 10, 6
Bin 3: 7, 11, 7

Hard practical problems

Often seem to need exponential time to solve
  • We rapidly become unable to solve them as problem instances become bigger
Why so hard

In the worst case, as far as we know, you have to systematically explore an exponential set of alternatives in order to find the exact solution.

Analyzing heuristics

Suppose the optimal is N.
Can FF exceed 2N?

A. Yes
B. No
**Algorithms**

Does value V occur in sorted list L?

---

**Simple scan**

set found = false

set i = 1

repeat until found or i > length of list

• if item i of list = value
  • set found = true

• else
  • change i by 1
What if never found?

• set found = false
• set i = 1
repeat until found or i > length of list
  – if item i of list = value
    • set found = true
  – else
    • change i by 1

How many times does this loop run if list contains N things?

A. 1
B. 10
C. N
D. $N^2$
E. N!

Better algorithm

Keep track of interval where item might be
Test the middle of the interval
Cut the interval in half if not found
**Pseudocode...**

set low = 1
set high = length of list
set pivot = ⌊(low + high)/2⌋

**Visualization**

```
  |   |
  |   |
  |   |
```

Low   Pivot   High

**Pseudocode...**

repeat until low > high (ie., not found)
or value = item pivot of list (ie., found)

**Visualization**

```
  |   |
  |   |
  |   |
```

Low   Pivot   High
**Pseudocode...**

- if value > item pivot of list
  set low to pivot + 1

**Visualization**

```
<table>
<thead>
<tr>
<th>Pivot</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td></td>
</tr>
</tbody>
</table>
```

**Pseudocode...**

- else (ie., value < item pivot of list)
  set high to pivot - 1

**Visualization**

```
| Low   | Pivot | High |
```


Pseudocode...

- finally set pivot = \(\lceil (\text{low} + \text{high})/2 \rceil\)

Visualization

\[
\begin{array}{c}
\text{Low} & \text{Pivot} & \text{High} \\
\end{array}
\]

Summary

set low = 1
set high = length of list
set pivot = \(\lceil (\text{low} + \text{high})/2 \rceil\)
repeat until low > high or value = item pivot of list
  - if value > item pivot of list
    set low to pivot + 1
  - else
    set high to pivot – 1
- set pivot = \(\lceil (\text{low} + \text{high})/2 \rceil\)