Outline

1. Example
2. Sorting review
3. Another Example
4. Hashing
5. One More Example
6. Third assignment
When presenting proofs, use some words like “since” and “therefore”. The three dots are forbidden.
Administrative Comments

- When presenting proofs, use some words like “since” and “therefore”. The three dots are forbidden.
- Answers without explicit justifications give 0 pts in an exam.
Administrative Comments

- When presenting proofs, use some words like “since” and “therefore”. The three dots are forbidden.
- Answers without explicit justifications give 0 pts in an exam.
- Numerical justification is no justification.
Outline

1 Example

2 Sorting review

3 Another Example

4 Hashing

5 One More Example

6 Third assignment
Example exam question

Prove or disprove:

a) \( \sqrt{n} = \Omega((\lg n)^2) \)

b) \( 2^n + (-2)^n + 1 = \Theta(2^n) \)

c) Let \( T(n) = \begin{cases} 1 & n = 1 \\ T(\sqrt{n}) + 1 & n > 1 \end{cases} \)

Is \( T(n) = \mathcal{O}(\lg \lg n) \)?

d) \( 2^{n+1} = \mathcal{O}(2^n) \)
Example exam question (contd)

Prove or disprove:

e) \( \lg(f(n)) = \mathcal{O}(\lg(g(n))) \implies f(n) = O(g(n)) \)

f) \( \max(f(n), g(n)) = \Theta(f(n) + g(n)) \)
Outline

1. Example
2. Sorting review
3. Another Example
4. Hashing
5. One More Example
6. Third assignment
Insertion Sort

- Simple
- In place sorting algorithm
- $O(n^2)$ worst case behaviour
Merge Sort

- Does not sort in place
- $\Theta(n \log n)$ worst case behaviour
Heapsort

- Relies in the heap data structure
- Max-Heapify is used to “bubble” up the values
- Sorts in place
- $\Theta(n \lg n)$ worst case behaviour
Quicksort

- Very common in practice
- Works with a pivot element and partitioning
- $\Theta(n^2)$ worst case behaviour
- $O(n \lg n)$ average case behaviour
Counting Sort

- Assumes all keys are in the interval $[0, k)$
- Works by counting the number of occurrences of each key
- $\Theta(n + k)$ worst case
Radix Sort

- Sorts a “column” at a time using a stable sorting algorithm
- $\mathcal{O}(d(n + k))$ to sort $n$ $d$-digit numbers in the worst case
Bucket sort

- Assumes input is uniformly distributed in [0, 1)
- Works by dividing input into buckets
- Expected running time $\Theta(n)$
Outline

1. Example
2. Sorting review
3. Another Example
4. Hashing
5. One More Example
6. Third assignment
Example exam question

Binary-Search\((A, \textit{start}, \textit{stop}, \textit{key})\)

1. ▶ Calculate the middle of the search-block
2. \( i \leftarrow \text{start} + \lceil (\text{stop} - \text{start})/2 \rceil \)
3. ▶ If we have failed in our search, we return a negative index
4. \textbf{if} \( \text{start} = \text{stop} \) \textbf{and} \( A[i] \neq \text{key} \)
5. \hspace{1em} \textbf{then} return \(-1\)
6. ▶ If we have found the key, we return its index
7. \textbf{if} \( A[i] = \text{key} \)
8. \hspace{1em} \textbf{then} return \(i\)
9. ▶ Otherwise, we recurse in the correct part of the array
10. \textbf{if} \( A[i] > \text{key} \)
11. \hspace{1em} \textbf{then} Binary-Search\((A, \textit{start}, i - 1, \textit{key})\)
12. \hspace{1em} \textbf{else} Binary-Search\((A, i + 1, \textit{stop}, \textit{key})\)

Derive the recurrence and its upper bound.
Outline

1. Example
2. Sorting review
3. Another Example
4. Hashing
5. One More Example
6. Third assignment
Outline

1. Example
2. Sorting review
3. Another Example
4. Hashing
5. One More Example
6. Third assignment
Problem

(15p)

Assume that we a set of four digit numbers that we want to translate into integers 0, ..., 9:

1066 1789 1945 1600 1915 2005 1000

Consider two hash functions $hashCode_1(x) = x \mod 10$ and $hashCode_2(x) = \frac{x - (x \mod 1000)}{1000}$. Assume numbers arrive from left to right.

a) Draw the resulting hash table if we use $hashCode_1$ and linear probing to resolve collisions.

b) Draw the resulting hash table if we use $hashCode_2$ and chaining to resolve collisions.

c) With the additional knowledge that the input numbers are all years, which of the two hash functions would be the better choice for arbitrary input?
Outline

1. Example
2. Sorting review
3. Another Example
4. Hashing
5. One More Example
6. Third assignment
Assignment 3.1

(i) Which of the following sorting algorithms are stable: insertion sort, merge sort, heapsort, and quicksort?

(ii) Give a simple scheme which makes any sorting algorithm stable. How much additional time is needed if your scheme is used?
Assignment 3.2

(i) Complete the following table:

<table>
<thead>
<tr>
<th>Data structure</th>
<th>Insert</th>
<th>Search</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linked List</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hash Table</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Search Tree</td>
<td></td>
<td></td>
<td>$\mathcal{O}(\lg n)$</td>
</tr>
</tbody>
</table>

Make sure you fill in expected time, and not worst case. State any assumptions you make, if anything is unclear.

(ii) With the help of your newly created table, describe a scenario for each data structure, where that particular data structure would be the best choice of the three. Justify your answer, and relate it to the table.

(iii) Do (i) also for worst case, in a new table.