Asymptotic Analysis

The Gist

Design and Analysis of Algorithms I

Motivation

Importance: Vocabulary for the design and analysis of algorithms (e.g. "big-Oh" notation).

- "Sweet spot" for high-level reasoning about algorithms.
- Coarse enough to suppress architecture/language/compiler-dependent details.
- Sharp enough to make useful comparisons between different algorithms, especially on large inputs (e.g. sorting or integer multiplication).

Asymptotic Analysis

High-level idea: Suppress constant factors and lower-order terms

too system-dependent

irrelevant for large inputs

Example: Equate $6n \log_2 n + 6$ with just $n \log n$.

Terminology: Running time is $O(n \log n)$

["big-Oh" of $n \log n$]

where $n =$ input size (e.g. length of input array).

Example: One Loop

Problem: Does array A contain the integer t? Given A (array of length n) and t (an integer).

Algorithm 1

```
for i = 1 to n do
    if A[i] == t then
        Return TRUE
Return FALSE
```

Question: What is the running time?

A) $O(1)$ C) $O(n)$

B) $O(\log n)$ D) $O(n^2)$

Example: Two Loops

Given A, B (arrays of length n) and t (an integer). [Does A or B contain t?]

Algorithm 2

```
for i = 1 to n do
    if A[i] == t then
        Return TRUE
for i = 1 to n do
    if B[i] == t then
        Return TRUE
Return FALSE
```

Question: What is the running time?

A) $O(1)$ C) $O(n)$

B) $O(\log n)$ D) $O(n^2)$

Example: Two Nested Loops

Problem: Do arrays A, B have a number in common? Given arrays A, B of length n.

Algorithm 3

```
for i = 1 to n do
    for j = 1 to n do
        if A[i] == B[j] then
            Return TRUE
Return FALSE
```

Question: What is the running time?

A) $O(1)$ C) $O(n)$

B) $O(\log n)$ D) $O(n^2)$

Example: Two Nested Loops (II)

Problem: Does array A have duplicate entries? Given arrays A of length n.

Algorithm 4

```
1: for i = 1 to n do
2:     for j = i+1 to n do
3:         if A[i] == A[j] then
4:             Return TRUE
5: Return FALSE
```

Question: What is the running time?

A) $O(1)$ C) $O(n)$

B) $O(\log n)$ D) $O(n^2)$