CSE 101 – Nov 4, 2019 (Week 6)

Notes provided by Ben Sihota bsihota@ucsc.edu

Example

Recall: MergeSort(A, p, r)

1. if $p < r$
   2. $q = \lfloor \frac{p+r}{2} \rfloor$
   3. MergeSort(A, p, q)
   4. MergeSort$(A, q+1, r)$
   5. Merge(A, p, q, r)

Operation	Cost
1	0
2	0
3	$T(\lceil \frac{n}{2} \rceil)$
4	$T(\lfloor \frac{n}{2} \rfloor)$
5	$n-1$

$$T(n) = \begin{cases} 0 & n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + (n-1) & n \geq 2 \end{cases}$$

Simplify (tight asymptotic bound):

$T(n) = 2T(\frac{n}{2}) + n$

Apply Master Theorem **Compare**: n to $n^{\log_2 2} = n^1$

Case 2: $T(n) = \Theta(n \log n)$

Example

Write 3-way MergeSort

$T(n) = 3T(\frac{n}{3}) + n$

Compare: n to $n^{\log_3 3} = n^1$

Case 2: $T(n) = \Theta(n \log n)$

Example

Find number of inversions in $A[1...n]$

Inversions(A, p, n)

1. if $p < r$
   2. $q = \lfloor \frac{p+r}{2} \rfloor$
   3. $a =$ Inversions(A, p, q)
   4. $b =$ Inversions$(A, q+1, r)$
   5. $c =$ Compare(A, p, q, r)
   6. return $a + b + c$
2. else
3. return 0

Compare(A, p, q, r)

1. count $= 0$
2. for $i = p$ to q
   3. for $j = q + 1$ to r
      4. if $A_i > A_j$
         5. count++
3. return count

Note: Compare$(A, 1, \lfloor \frac{n+1}{2} \rfloor, n)$

Runs in time: $\Theta(\lceil \frac{n}{2} \rceil * \lfloor \frac{n}{2} \rfloor) = \Theta(n^2)$

So if $T(n) =$ number comparisons in Inversions$(A, 1, n)$, then

$$T(n) = \begin{cases} 0 & n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + \lceil \frac{n}{2} \rceil * \lfloor \frac{n}{2} \rfloor & n \geq 1 \end{cases}$$

Simplify:

$T(n) = 2T(\frac{n}{2}) + n^2$

Compare: n^2 to $n^{\log_2 2} = n^1$

Check regularity condition

Case 3: $T(n) = \Theta(n^2)$

Graph Theory Handout

Example

$V_1 = \{1, 2, 3, 4, 5, 6\}$

$E_2 = \{12, 14, 23, 24, 25, 26, 35, 36, 45, 56\}$

Example

$V_2 = \{a, b, c, d, e, f\}$

$E_2 = \{ab, ad, bc, bd, bf, ce, cf, fe, be, de\}$

Notice $\phi : V_1 \to V_2$ (bijection)

$1 \to a$

$2 \to b$

$3 \to c$

$4 \to d$

$5 \to e$

$6 \to f$

Has property: $\{x, y\} \in E, \iff \{\phi(x), \phi(y)\} \in E_2$

We call such a mapping $\phi : V_1 \check{} V_2$ an **isomorphic**. We say G_1 and G_2 are **isomorphic**, write $G_1 \cong G_2$.

Definition

Let $x, y \in V(G)$

- $x - y$ **walk** in G:
  - $x = v_0, v_1, ..., v_k = y$ (length $= k$) such that $\{v_{i-1}, v_i\} \in E(G)$ for $i = 1, ..., k$
  - Walk is **closed** if $x = y$
  - Trivial walk: $v_0 = x = y$
- $x \check{} y$ **trail** in G:
  - A walk in which no edge is traversed more than once
- $x \check{} y$ **path** in G:
  - An $x \check{} y$ trail in which no vertex is repeated (unless $x = y$)
- A closed, non-trivial path is a **cycle**

Definition

Given $x, y \in V(G)$

$$D(x,y) = \begin{cases} \text{length of a min-length } x \breve{\ } y \text{ path, if such a path exists} \\ \infty \text{ if no } x \breve{\ } y \text{ path exists} \end{cases}$$

Note: A minimum length $x\breve{\ }y$ walk is necessarily a path

Problem

Single-source shortest path (SSSP): Given any $s\ inV(G)$

(1) Determine $D(s,x)$ for all $x \in V(G)$

(2) For each x that is reachable from s, find a shortest $s\breve{\ }x$ path