COP 3503 11/3/26

Divide + Conquer

- Integer Mult / Polynomial Mult

  Alg $O(n^{\log_2 3})$ run-time

  Rec Rel $T(n) = 3T\left(\frac{n}{2}\right) + O(n)$

  ↑ # rec calls; ↑ size input for each; ↑ extra work adding stuff

- Tromino Tiling

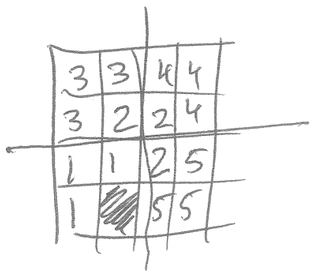
- Pair of Closest Points

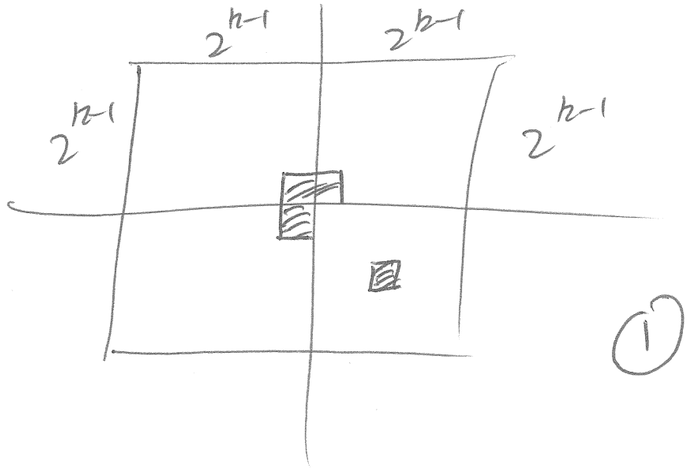
- Function Composition (Shuffling) — UCF HS Contest

Tromino Tiling



Goal: Given $2^k \times 2^k$ grid w/ 1 unit sq missing





Idea: Split Tiling into four $2^{k-1} \times 2^{k-1}$ areas (quadrants)

(1) Recursively Tile Quadrant w/ missing tile

(2) Put a tile in center in other 3 quadrants

(3) Recursively Tile other 3 quadrants

$n = 2.5 \times 10^5 \rightarrow 4.3$ sec

$n = 10^5 \rightarrow$

$T(n) = c \cdot n^{\log_2 3}$

$T(2.5 \times 10^5) = c \cdot (2.5 \times 10^5)^{\log_2 3} = 4.3$ sec

$$c = \frac{4.3 \text{ sec}}{(2.5 \times 10^5)^{\log_2 3}}$$

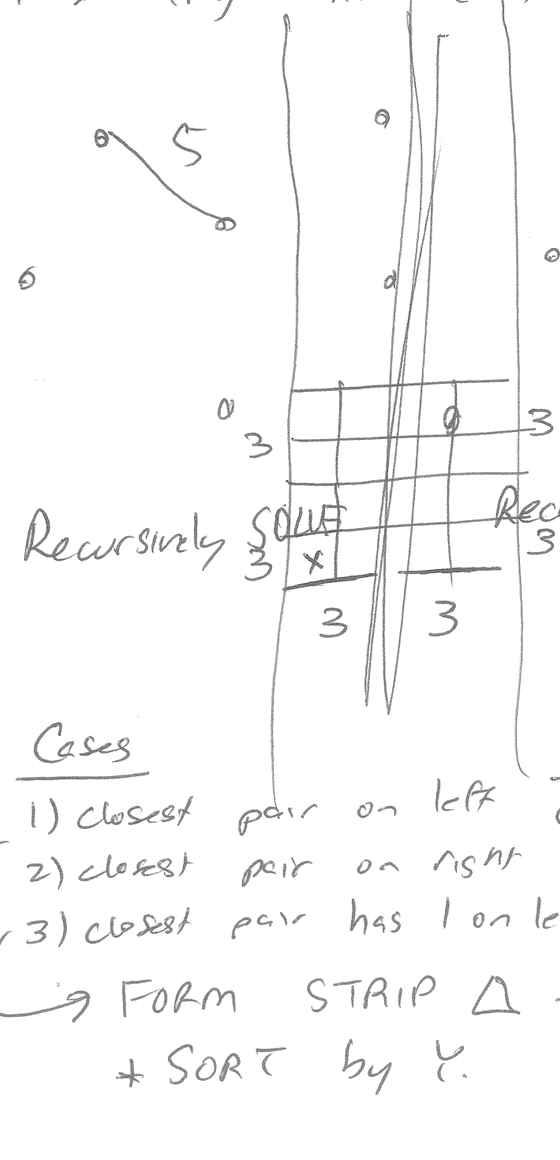
$$T(10^5) = \frac{4.3 \; (10^5)^{\log_2 3}}{(2.5^{\log_2 3})(10^5)^{\log_2 3}}$$

Pair of Closest Points

Input: n pts Cartesian Plane

Output: Distance btw pair of closest pts
OR Pr 2 pts that are closest together.

BF: Try all $\binom{n}{2}$ pairs of pts



① Sort by x

② Divide pts by "median" x val

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

↑ # rec calls ↑ Size of each rec call

Cases

1) closest pair on left
2) closest pair on right
} $\min \Delta = \min(\text{left}, \text{right})$

3) closest pair has 1 on left, 1 on right
↳ FORM STRIP Δ from dividing line
+ SORT by y.

```
for (int i = 0; i < s2; i++)
    for (int j = i; j < min(i+8, s2); j++)
        if dist(i,j) < min
            blah blah
```

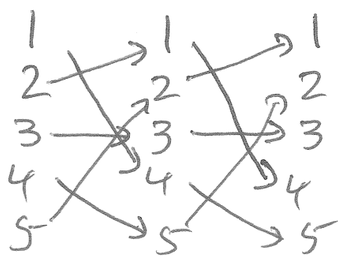
O(n) work!

Shuffling

Given a permutation function on input 1..n and an integer k, determine $f^k(x)$ - composing f with itself k times

f

x	f(x)
1	4
2	1
3	3
4	5
5	2



$f^{1000000}(x)$

$= f^{500000}(f^{500000}(x))$

1 → 5
2 → 4
3 → 3
4 → 2
5 → 1

Composed ~~to~~ f(f(x)).

```
compute(int[] f, int k) {
    if (k == 1) return f;
    if (k%2 == 0) {
        int[] tmp = compute(f, k/2);
        return compose(tmp, tmp);
    }
    int[] tmp = compute(f, k-1);
    return compose(tmp, f);
}
```

Runtime = $O(n \log k)$

- n: perm size
- $\log k$: # compositions

GRAPHS

Divide+Conquer
- Trominos
- Int Mult
- Closest Pair Pts
- Function Composition

- Store
- DFS, BFS
- TopSort
- BFS applications
- Weighted
- Dijkstra's
- Bellman Ford
- Floyd Warshalls

SHORTEST DIST

MST
- Kruskal's
- Prims

Network Flow
- emphasize graph formation not N.F. code

+1 SUMMATION