CSCI 270

Discussion Week 6

Implementation of closest pair points

Driver :

closest_pair(p):

Construct P_x : list of of points sorted by x-coord

Construct P_y : list of of points sorted by y-coord

(p_0, p_1) = closest_pair_rec(P_x, P_y)

$\Theta(n\log n)$

```
closest_pair_rec(P_x, P_y):
    if |P| ≤3:
        solve directly
    else
        Construct Q_x, left half of P_x
        Construct Q_y, corresponding points sorted by y-coord
        Construct R_x, right half of P_x
        Construct R_y, corresponding points sorted by y-coord
```

closest_pair_rec(P_x, P_y):

if $|P| \leq 3$:

solve directly

else

Construct Q_x, left half of P_x — $\Theta(1)$

Construct Q_y, corresponding points sorted by y-coord — $\Theta(n)$

Construct R_x, right half of P_x — $\Theta(1)$

Construct R_y, corresponding points sorted by y-coord — $\Theta(n)$

Divide

(q_0, q_1) = closest_pair_rec(Q_x, Q_y)

(r_0, r_1) = closest_pair_rec(R_x, R_y)

X = min((d(q_0, q_1), d(r_0, r_1))

S = set of points within a distance of X from L

Construct S_y - S sorted by y-coord

$\Theta(n)$

for each point in S_y, compute its distance to each of the next 11 points in S_y.

$\Theta(n)$

let (s, s’) be the pair with closest distance

if d(s, s’) < X:

    return (s, s’)

elif $d(q_0, q_1) < d(r_0, r_1)$:

    return (q_0, q_1)

else:

    return (r_0, r_1)

Applying Master's Theorem

$a = 2, b = 2$

$n^{\log_2 2} = n$

$f(n) = \Theta(n)$

Driver: $\Theta(n\log n)$

Case #2 -> $\Theta(n\log n)$

Problem

Solve the following recurrences using the Master Method:

a) $A(n) = 3\, A(n/3) + 15$
b) $B(n) = 4\, B(n/2) + n^3$
c) $C(n) = 4\, C(n/2) + n^2$
d) $D(n) = 4D(n/2) + n$

Solution

a) $f(n) = 15 = O(1)$, $n^{\log_b a} = n^{\log_3 3} = n^1$ This falls under case 1 $A(n) = \theta(n)$

b) $f(n)=n^3$, $n^{\log_b a} = n^{\log_2 4} = n^2$

This can fall under case 3. Now we need to check that

$af(n/b) \leq cf(n)$ for some $c<1$:

$a\, f(n/b) = 4 * (n/2)^3 = 4 * n^3/8 = n^3/2 = .5\, f(n)$, so we have found $c=.5$ such that

$a\, f(n/b) \leq c\, f(n)$ and the inequality checks out, and case 3 applies $B(n) = \theta(n^3)$

c) $f(n)=n^2$, $n^{\log_b a} = n^{\log_2 4} = n^2$

This falls under case 2 : $C(n) = \theta(n^2 \log n)$

d) $f(n)=n$, $n^{\log_b a} = n^{\log_2 4} = n^2$

This falls under case 1 : $D(n) = \theta(n^2)$

Problem

There are 2 sorted arrays A and B of size n each. Design a D&C algorithm to find the median of the array obtained after merging the above 2 arrays (i.e. array of length 2n). Discuss its runtime complexity.

Solution

Find the median of the two arrays. Say the medians are mA and mB. If mA=mB, then this is our median.

Otherwise, say mA < mB then throw away all terms lower than mA in A, and all terms greater than mB in B. Solve the resulting subproblem recursively.

Complexity analysis:

- Divide step takes O(1). This includes finding medians in A and B and throwing away half of A and B.
- There is no combine step
- Number of subproblems (a) at each step is 1. The size of the subproblem (n/b) is n/2

So we can apply the Master Method:

$f(n) = O(1)$, $n^{\log_b a} = n^{\log_2 1} = n^0 = O(1)$

This falls under case 2 $T(n) = \theta(\log n)$

Problem

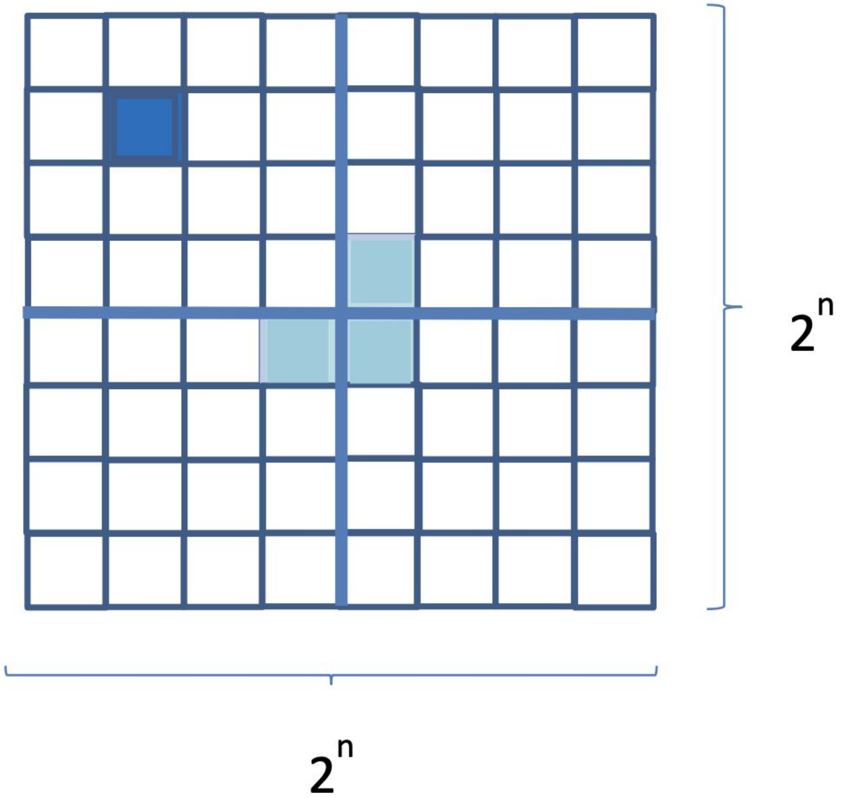
A tromino is a figure composed of three 1x1 squares in the shape of an L. Given a 2^nx2^n checkerboard with 1 missing square, tile it with trominoes. Design a D&C algorithm and discuss its runtime complexity.

Solution

The figure below shows a 2^nx2^n checkerboard with a hole highlighted in dark blue.
Here is a divide and conquer solution:

- Divide the grid into 4 equal grids of size 2^{n-1}x2^{n-1}
- Add holes to the three grids that do not have a hole in them. The position of the new holes should be at the center of the grid so that when each region is solved/tiled recursively we can cover the remaining three holes with one tromino.
- Solve all 4 subproblems recursively
- When combining the solutions, place a tile over the three holes in the center to complete the tiling.
- During recursion, when we reach 2x2 grids, we can solve the problem directly by placing a single tromino to tile the grid (that has a single hole).

Solution



Solution

Complexity analysis:

Divide steps takes O(1) time. So does the combine step.
Number of subproblems (a) = 4, size of each subproblem is half the size of the original problem, so b=2.

Let $x = 2^n$

$f(x)=O(1)$, $x^{\log_b a} = x^{\log_2 4} = x^2$ This falls under case 1 : $T(x) = \theta(x^2)$

Problem

1. Suppose we have two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, along with T_1 which is a MST of G_1 and T_2 which is a MST of G_2. Now consider a new graph $G = (V, E)$ such that $V = V_1 \cup V_2$ and $E = E_1 \cup E_2 \cup E_3$ where E_3 is a new set of edges that all cross the cut (V_1, V_2).

Consider the following algorithm, which is intended to find a MST of G.

```
Maybe-MST(T₁, T₂, E₃)
        e_min = a minimum weight edge in E₃
        T = T₁ U T₂ U { e_min}
        return T
```

Does this algorithm correctly find a MST of G? Either prove it does or prove it does not.

Solution

No. Counterexample ->

The correct solution will have edges AB, DC, and AD in the MST. The algorithm will result in an incorrect MST with edges AD, BC, and DC.