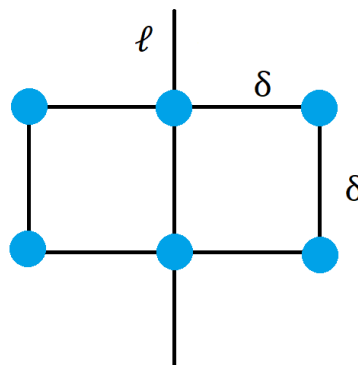


Họ tên: Nguyễn Trung Dũng

MSSV: 19120486

Bài 1:

- Ta có: $\delta = \min\{\delta_L, \delta_R\}$ với δ_L, δ_R lần lượt là khoảng cách nhỏ nhất giữa 2 điểm bất kì ở bên trái và bên phải
 \Rightarrow Một cặp điểm bất kì ở cùng một phía của ℓ luôn có khoảng cách $d \geq \delta$
- Giả sử $\delta_L = \delta_R = \delta$
 \Rightarrow Cách phân bố sao cho có tối đa số điểm trong hình chữ nhật $2\delta \times \delta$ là các điểm đó phải nằm trên các điểm góc như hình sau:



Vậy tồn tại tối đa 6 điểm trong vùng tìm kiếm diện tích $2\delta \times \delta$

Bài 2:

- Thao tác cơ sở:
`tmpSum = moneyChange(coins, i) + moneyChange(coins, money - i);`
 \Rightarrow Hệ thức truy hồi:

$$T(n) = \begin{cases} \sum_{i=1}^{\lfloor n/2 \rfloor} (T(i) + T(n-i)) + \Theta\left(\frac{n}{2}\right) & n > 2 \\ 1 & n = 2 \\ 0 & n = 1 \end{cases}$$
$$\Rightarrow T(n) = \begin{cases} \sum_{i=1}^{n-1} T(i) + \Theta\left(\frac{n}{2}\right) & n > 2 \\ 1 & n = 2 \\ 0 & n = 1 \end{cases}$$

- Có:

$$T(n) = \sum_{i=1}^{n-1} T(i) + \Theta\left(\frac{n}{2}\right) \quad (1)$$

$$T(n+1) = \sum_{i=1}^n T(i) + \Theta\left(\frac{n+1}{2}\right) \quad (2)$$

Lấy (2) – (1):

$$T(n+1) - T(n) = T(n) + \Theta\left(\frac{n+1}{2}\right) - \Theta\left(\frac{n}{2}\right)$$

$$\Leftrightarrow T(n+1) = 2T(n) + \Theta\left(\frac{n+1}{2}\right) - \Theta\left(\frac{n}{2}\right)$$

$$\Rightarrow T(n) = 2T(n-1) + \Theta\left(\frac{n}{2}\right) - \Theta\left(\frac{n-1}{2}\right)$$

Có thể thấy rằng $\Theta\left(\frac{n}{2}\right)$ và $\Theta\left(\frac{n-1}{2}\right)$ có cùng bậc (bằng 1) và cùng hệ số trước n

$$\Rightarrow \Theta\left(\frac{n}{2}\right) - \Theta\left(\frac{n-1}{2}\right) \approx \Theta(1)$$

$$\Rightarrow T(n) = 2T(n-1) + c \quad (c \text{ là hằng số})$$

$$= 2^{n-2}T(2) + c(1 + 2 + 4 + \dots + 2^{n-3})$$

$$= 2^{n-2} + c \frac{1 - 2^{n-2}}{1 - 2}$$

$$= (c+1)2^{n-2} - c$$

$$= \frac{c+1}{4} 2^n - c \in \Theta(2^n)$$

Bài 3:

- Chương trình nhân 2 số nguyên dương n bit (C++):

```
// tìm bit cao nhất (most significant bit)
int msb(int a) {
    if(a == 0)
        return 0;

    // giảm a cho tới khi chỉ còn lại bit cao nhất
    while(a - (a&-a))
        a -= a&-a;
```

```

    // tìm vị trí của bit cao nhất
    int i = 0;
    while((a & (1<<i)) == 0)
        i++;
    return i+1;
}

int Multiplication(int x, int y) {
    int n = max(msb(x), msb(y)); // tìm bit cao nhất
    if (x == 0 || y == 0)
        return 0;
    if (n <= 4) // nếu bit cao nhất <= 4 thì dùng built-in operator
        return x*y;

    n += n&1; // nếu n lẻ thì cho n += 1, nếu không thì n += 0
    int m = n/2;

    int xl = x >> m;
    int xr = x & ((1<<m) - 1);
    int yl = y >> m;
    int yr = y & ((1<<m) - 1);

    int ll = Multiplication(xl, yl); // tính xl*yl
    int rr = Multiplication(xr, yr); // tính xr*yr
    int lr = Multiplication(xl + xr, yl + yr) - (ll + rr); // tính xl*yr + xr*yl

    int ans = (ll << n) + (lr << m) + rr;
    return ans;
}

```