

List-Heap

Del key / Add / Merge / Min $O(\log n)$

Ext Min $O(n)$



Def. Binomial Tree

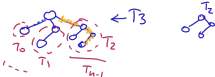
$T_0 = \bigcirc$

T_1 :

T_2 :

T_n :





YTC: $|T_n| = 2^n$

$$|T_n| = 1 + \underbrace{|T_0|}_{2^0} + \underbrace{|T_1|}_{2^1} + \dots + |T_{n-1}| = 2^n$$

YTC $\text{depth}(T_n) = n$

$\text{depth}(M) = O(\log n)$

YTC $T_n = T_{n-1} + \hat{T}_{n-1}$

```
struct Node {  
    int key;  
    int rank;
```

```
(*) Node* parent;  
    Node* child;  
    Node* next;  
    Node* prev;  
};
```



Binomial Heap: {List of T_i

$$n = \sum_i z^k: |T_i| = z^i \quad \text{Min}$$

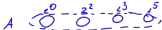
$$6 = 2^1 + 2^2$$

height $\leq \log_2 n$

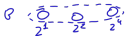


- ① Add \leftarrow Merge $O(\log n)$
- ② Merge $O(\log n)$
- Min : $O(\log n)$ $O(1)$
- ExtMin \leftarrow Merge $O(\log n)$
- DecreaseKey \leftarrow siftup $O(\log n)$
- Delete \leftarrow DecreaseKey $O(\log n)$





$$T_{h+1} \leftarrow T_h + T_h$$



Normalize (List of Pools: lst) \rightarrow Binom. Heap

Result = [None, ...,] $\approx \log$

for v in lst:

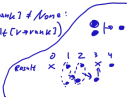
while Result[v.rank] \neq None:

$\quad v = \text{join}(v, \text{Result}[v.rank])$

Result[v.rank] = v

return Result

$O(|\text{lst}| \log n)$



Relaxed Binomial Heap: $\{L: \text{set}(Node^*) \mid \text{Node} = m \cdot 2^i\}$

$$6 = 2^1 + 2^2$$



Merge: $O(1)$



Add: $O(1)$

Min: $O(1)$

Searches: $O(\log n)$ ✓ $O(1)$, avg

Delete (avg) $O(\log n)$

ExtractMin (avg) $O(\log n)$

ExtMin: $|est| + \log$



Normalize: $|est| \rightarrow \log$

$$\varphi = |est|$$

$$\Delta\varphi = -|est| + \log$$

$$f = |est| + \log$$

$$a = \epsilon \Delta\varphi = O(\log)$$

Build Heap: $O(n)$

Fibonacci Heap (1984, Fredman, Turpin)



DecreaseKey: $\Theta(1)$



DecreaseKey(v)

```

if v → parent == null
  or not Bad(v → parent → key,
              v → key)
  return
  
```

Def cut(v):

v → parent → tag = 1

"If you v is global.

Change global v → parent"

(st.push, Bad(v); v → marked = false;

// v → parent → marked = 1;

if (v → parent → parent == null) return;

if (v → parent → marked) cut(v → parent);

v → parent → marked = 1;

cut(v)

v → key, tag, marked, 3:





Def $RANK(Root) = \text{deg Root}$

$$\begin{aligned} RANK(v) &= \\ &= \text{deg}(v) \\ &+ v \text{ marked} \end{aligned}$$



$$\psi = |tst| + 2 \text{Marked}$$

Diagram illustrating a tree structure. The root node is marked with a solid circle. Its children are marked with dashed circles. The root has a degree of k . The children have a degree of $k-1$. The root is marked, so its rank is k . The children are not marked, so their rank is $k-1$.

$$t = k+1$$

$$\Delta |tst| = k+1$$

$$\Delta \text{Marked} = -k$$

$$\Rightarrow a = \mathcal{O}(1)$$



$S_n = \text{min partitio}$ aggregata
 Adhuc h.

$$S_n = 1 + S_1 + \dots + S_{n-1}$$

$$S_{n-1} = 1 + S_1 + \dots + S_{n-2}$$



$$S_0 = 1$$

$$S_1 = 1$$

$$S_n = \underbrace{1 + S_0 + S_1 + \dots + S_{n-3} + S_{n-2}}_{S_{n-1}}$$

$$S_n = S_{n-1} + S_{n-2}$$

Prob.

$$f_0 = 1$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2}$$

$$\sim \left(\frac{1 + \sqrt{5}}{2} \right)^n$$

$$\sim \log_{\frac{1+\sqrt{5}}{2}}$$



-1 pabihok =>



join



recheigt

recheigt?

$\geq 0 \geq 1 \dots \geq r-1 \geq r$
 $\geq 0 \geq 1 \dots \geq r$

$\circ k \text{ or } k = k$

$0 \quad 0 \quad \dots \quad 0$
 $r_0 \quad r_1 \quad \dots \quad r_{k-1}$

$r_i: r_i + 1$

4/6: $r_i \geq 2$



$\geq 5_r$

$k \text{ or } k-1$

$1 + 0 \cdot 5_r + 1 \cdot 5_r + \dots$
 $\underbrace{\hspace{10em}}_{5_r}$

$\geq 0 \geq 1 \dots \geq k-1$

