18a-0: Binomial Trees

- $B_0$ is a tree containing a single node
- To build $B_k$:
  - Start with $B_{k-1}$
  - Add $B_{k-1}$ as left subtree

18a-1: Binomial Trees

\[ B_0 \quad B_1 \quad B_2 \quad B_3 \quad B_4 \]

18a-2: Binomial Trees

\[ B_0 \quad B_1 \quad B_2 \quad B_3 \quad B_4 \]

18a-3: Binomial Trees

- Equivalent definition
  - $B_0$ is a binomial heap with a single node
  - $B_k$ is a binomial heap with $k$ children:
    - $B_0 \ldots B_{k-1}$

18a-4: Binomial Trees

\[ B_0 \quad B_1 \quad B_2 \quad B_3 \quad B_4 \]
18a-5: **Binomial Trees**

B₀ B₁ B₂ B₃ B₄

18a-6: **Binomial Trees**

- Properties of binomial trees $B_k$
  - Contains $2^k$ nodes
  - Has height $k$
  - Contains $\binom{k}{i}$ nodes at depth $i$ for $i = 0 \ldots k$

18a-7: **Binomial Heaps**

- A Binomial Heap is:
  - Set of binomial trees, each of which has the heap property
    - Each node in every tree is $\leq$ all of its children
  - All trees in the set have a different root degree
    - Can’t have two $B_3$’s, for instance

18a-8: **Binomial Heaps**

```
  10
   /|
  22 7
   /|
  25 13 15
```

```
  5
   |
  12
   |
  13 15
   |
  17
```

```
  8
   |
  15
   |
  20
```

18a-9: **Binomial Heaps**

- Representing Binomial Heaps
  - Each node contains:
- left child, right sibling, parent pointers
- degree (is the tree rooted at this node $B_0$, $B_1$, etc.)
- data
- Each list of children sorted by degree

18a-10: **Binomial Heaps**

```
Head
10
  0
  5
  2
22
  1
  7
  0
12
  2
  9
  1
25
  0
  13
  1
17
  0
```

18a-11: **Binomial Heaps**

- How can we find the minimum element in a binomial heap?
- How long does it take?

18a-12: **Binomial Heaps**

- How can we find the minimum element in a binomial heap?
  - Look at the root of each tree in the list, find smallest value
- How long does it take?
  - Heap has $n$ elements
  - Represent $n$ as a binary number
  - $B_k$ is in heap iff $k$th binary digit of $n$ is 1
  - Number of trees in heap $\in O(\log n)$

18a-13: **Binomial Heaps**

- Merging Heaps $H_1$ and $H_2$
  - Merge root lists of $H_1$ and $H_2$
  - What property of binomial heaps may be broken?
  - How do we fix it?
18a-14: **Binomial Heaps**

- Merging Heaps $H_1$ and $H_2$
  - Merge root lists of $H_1$ and $H_2$
    - Could now have two trees with same degree
  - Go through list from smallest degree to largest degree
    - If two trees have same degree, combine them into one tree of larger degree
    - If three trees have same degree (how can this happen?) leave one, combine other two into tree of larger degree

18a-15: **Binomial Heaps**

```
10  5
  /|
  22 7 12 9 15
    |
    25 13 15 20
      |
      17
```

```
11  3
  /|
  14 6
    |
    30
```

18a-16: **Binomial Heaps**

```
10  11  5  3  8
  /|
  22 7 14 6 12 9 15
    |
    25 30 13 15 20
      |
      17
```

18a-17: **Binomial Heaps**

```
10  5  3  8
  /|
  11 22 7 14 6 12 9 15
    |
    25 30 13 15 20
      |
      17
```

18a-18: **Binomial Heaps**
18a-19: Binomial Heaps

- Removing minimum element
  - Find tree $T$ that has minimum value at root, remove $T$ from the list
  - Remove the root of $T$
    - Leaving a list of smaller trees
  - Reverse list of smaller trees
  - Merge two lists of trees together

18a-20: Binomial Heaps

- Removing minimum element
18a-22: Binomial Heaps

- Removing minimum element

18a-23: Binomial Heaps

- Removing minimum element

18a-24: Binomial Heaps
• Removing minimum element

18a-25: Binomial Heaps

• Removing minimum element

18a-26: Binomial Heaps

• Removing minimum element

18a-27: Binomial Heaps
• Removing minimum element

![Diagram of a binomial heap]

18a-28: **Binomial Heaps**

• Removing minimum element

• Time?

18a-29: **Binomial Heaps**

• Removing minimum element

• Time?
  • Find the smallest element: $O(\lg n)$
  • Reverse list of children $O(\lg n)$
  • Merge heaps $O(\lg n)$

18a-30: **Binomial Heaps**

• Decreasing the key of an element (assuming you have a pointer to it)

18a-31: **Binomial Heaps**

• Decreasing the key of an element (assuming you have a pointer to it)
- Decrease key value
- While value < parent, swap with parent
  - Exactly like standard, binary heaps
- Time: $O(\lg n)$

18a-32: **Binomial Heaps**

- How could we delete an arbitrary element (assuming we had a pointer to this element)?

```
        Delete
        this key
10           8
    /     \\    \\
  5       12  9  15
   | \\
  22    7  13  15  20
   | \\
  25    17
```

18a-33: **Binomial Heaps**

- How could we delete an arbitrary element (assuming we had a pointer to this element)?
  - Decrease key to $-\infty$, Time $O(\lg n)$
  - Remove smallest, Time $O(\lg n)$