

Banded DP

Y: ' ' O C U R R A N C E

X: ' ' O C C U R R E N

0	1	2	3	4	5	6	7	8	9
1	0	1	2	3	4	5	6	7	8
2	1	0	1	2	3	4	5	6	7
3	2	1	1	2	3	4	5	5	6
4	3	2	1	2	3	4	5	6	6
5	4	3	2	1	2	3	4	5	6
6	5	4	3	2	1	2	3	4	5
7	6	5	4	3	2	2	3	4	4
8	7	6	5	4	3	3	2	3	4

' ' O C U

0	1	2	3
1	0	1	2
2	1	0	1
3	2	1	1
4	3	2	2
5	4	3	2
6	5	4	3
7	6	5	4
8	7	6	5

$$P_1$$

$$P_2$$

$$P_3$$

$$P_4$$

$$P_5$$

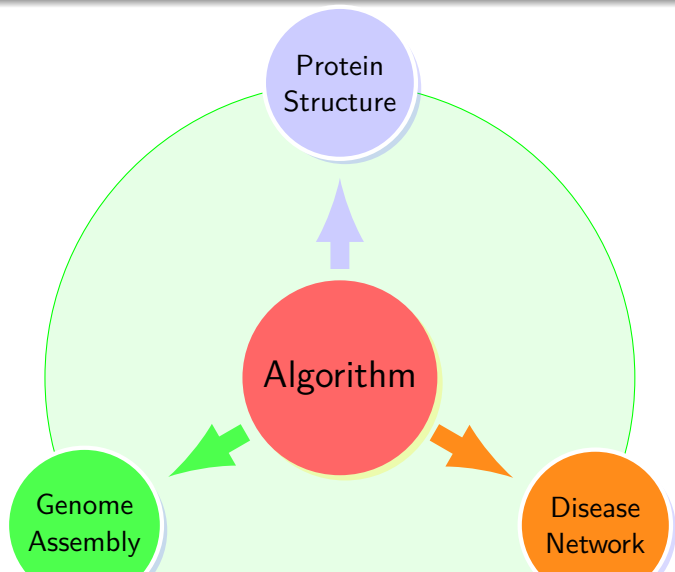
$$x_1 = 1$$

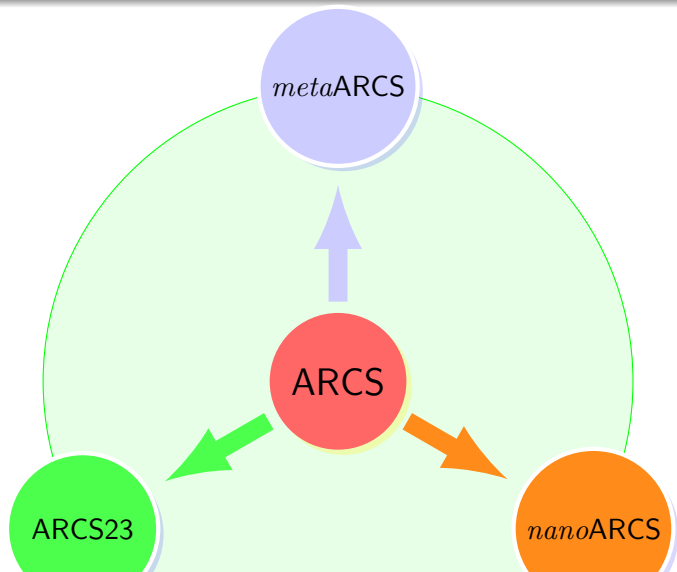
$$x_1 = 0$$

$$x_2 = 1$$

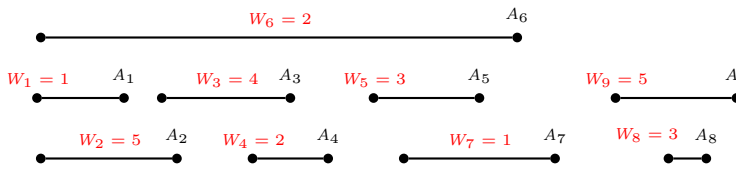
$$x_2 = 0$$

$$\{P_2, P_4, P_5\}$$

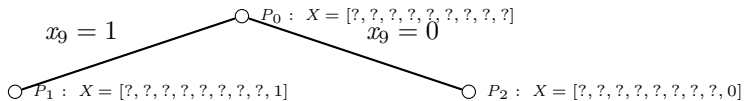




AAA



BBB



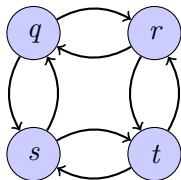
GREEDYINTERVALSCHEDULING(*CourseSet*)

- 1: **while** *CourseSet* $\neq \emptyset$ **do**
- 2: Select the course *C* with **earliest finishing time**;
- 3: Remove *C* and related courses from *CourseSet*;
- 4: **end while**

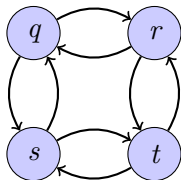
NNINTERVALSCHEDULING(*CourseSet*)

- 1: **while** *CourseSet* $\neq \emptyset$ **do**
- 2: Select the course *C* with **highest score by NN(*CourseSet*)**;
- 3: Remove *C* and related courses from

Coin



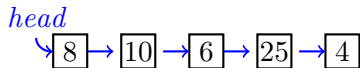
Longest path



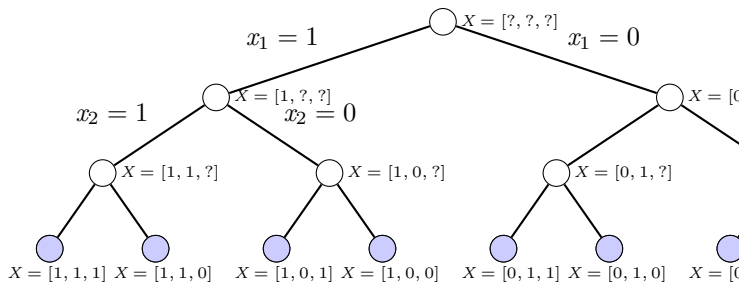
Lec7 Array

8	10	6	25	4			
---	----	---	----	---	--	--	--

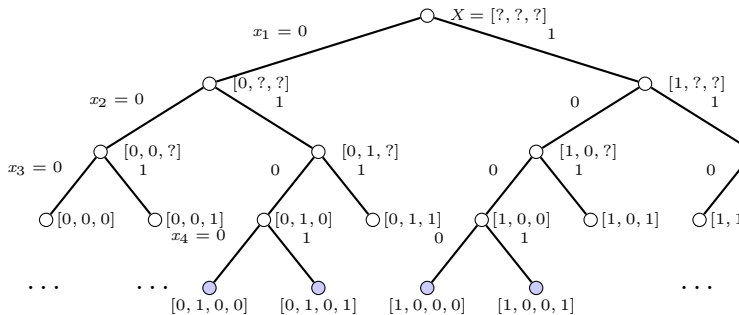
Lec7 Linked list



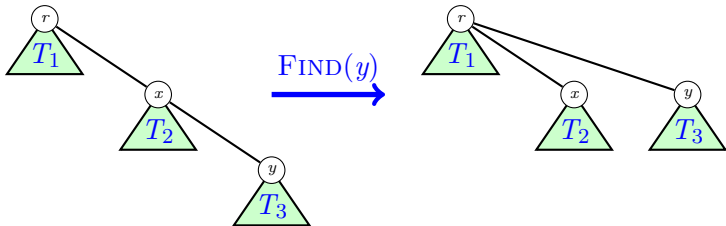
Lec1 Tree



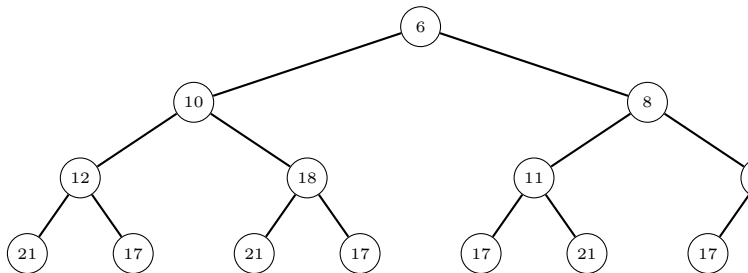
Tree



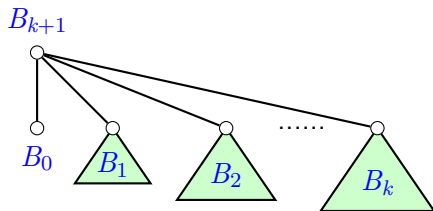
Union-Find path compression



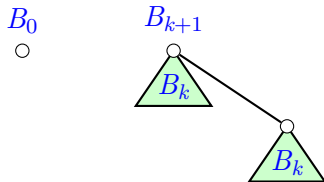
Lec6 DP 1



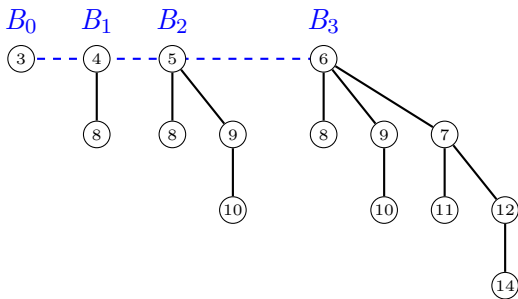
Binomial tree $B_k B_{k-1}$



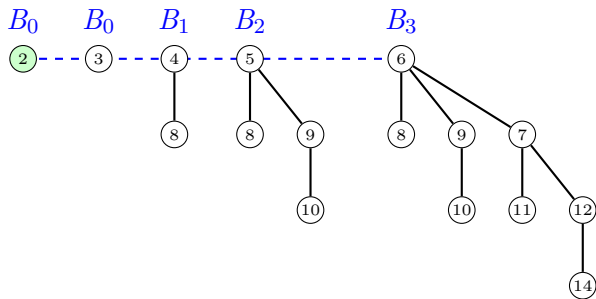
Binomial tree B_k



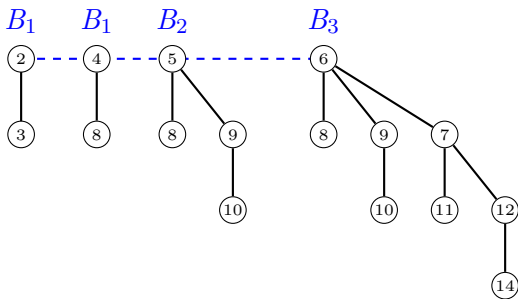
Binomial tree B_{0123} Insert 1



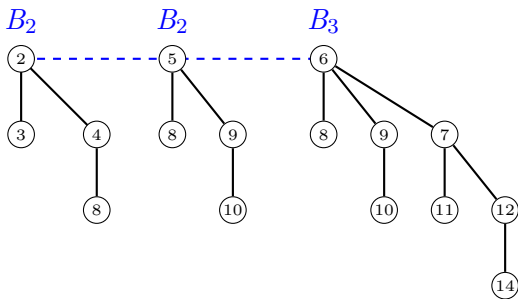
Binomial tree B_{0123} Insert 1



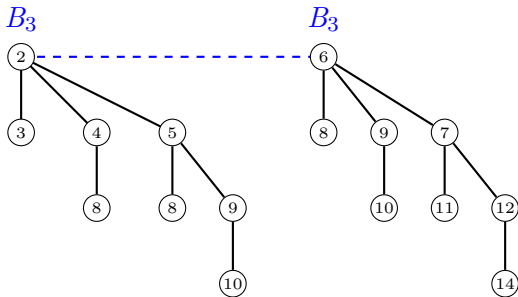
Binomial tree B_{0123} Insert 1 Step 2



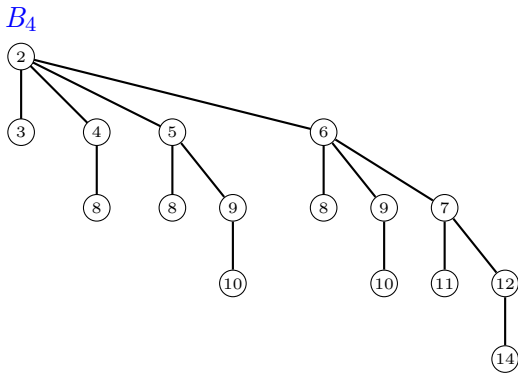
Binomial tree B_{0123} Insert 1 Step 3



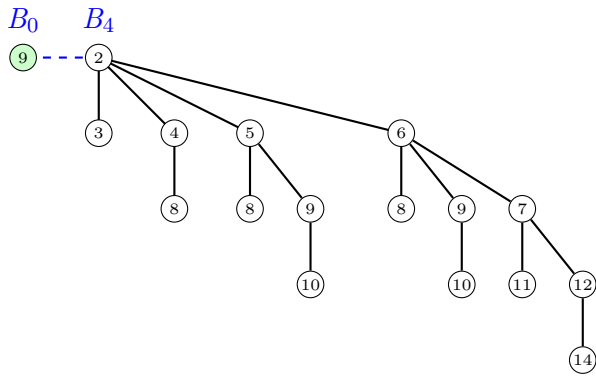
Binomial tree B_0123 Insert 1 Step 4



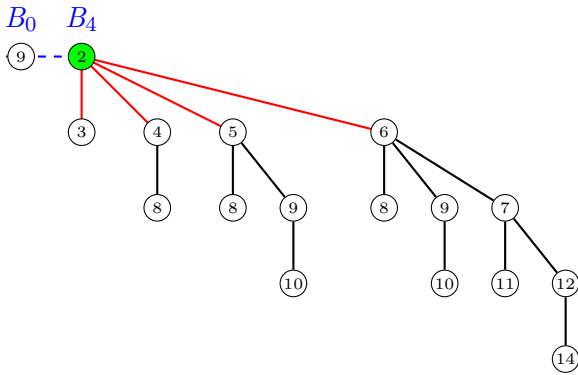
Binomial tree B_{0123} Insert 1 Step 5



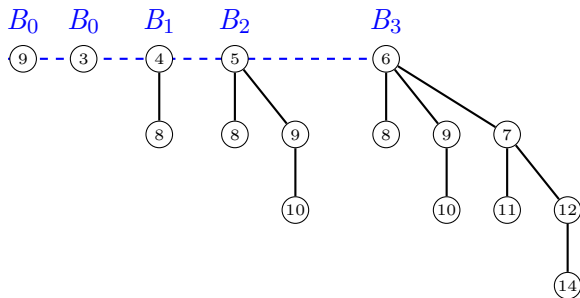
Binomial tree B_{0123} Insert 9 again



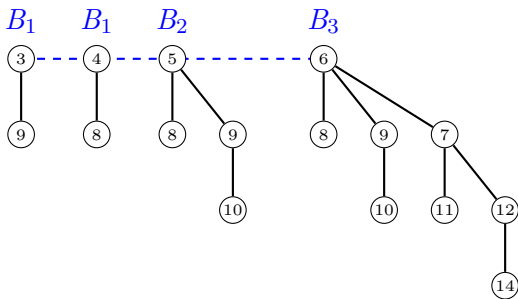
Binomial tree B_{0123} ExtractMin



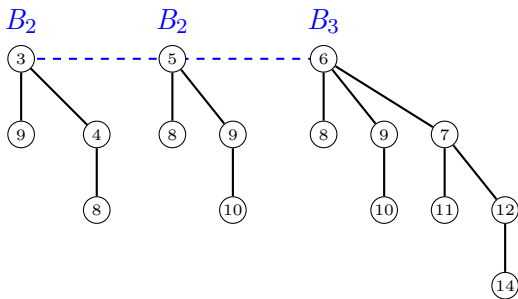
Binomial tree B_{0123} ExtractMin



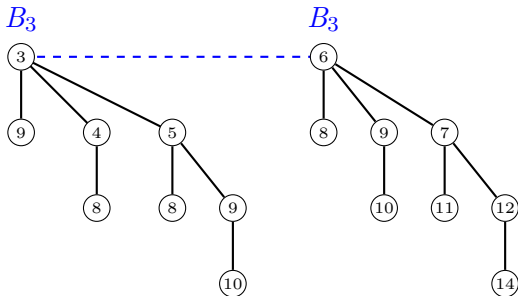
Binomial tree B_{0123} ExtractMin



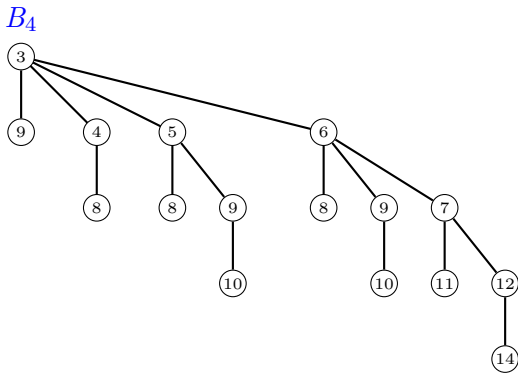
Binomial tree B_{0123} ExtractMin



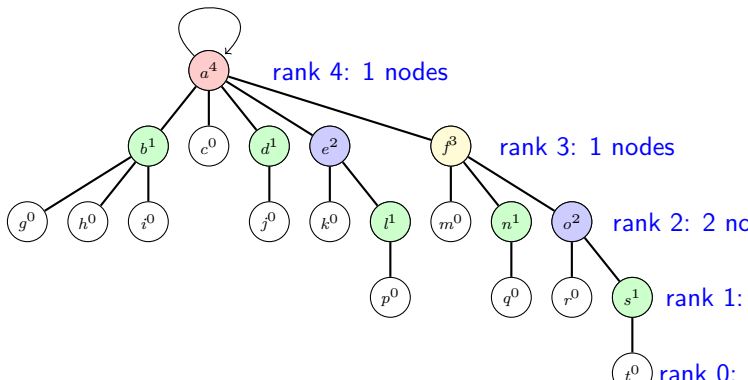
Binomial tree B_0123 ExtractMin



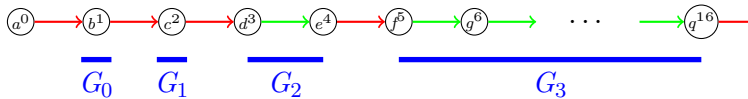
Binomial tree B_0123 ExtractMin



Union-Find tree



Links with group id



Binomial tree B_0

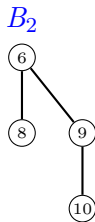
B_0

6

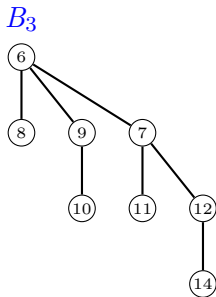
Binomial tree B1



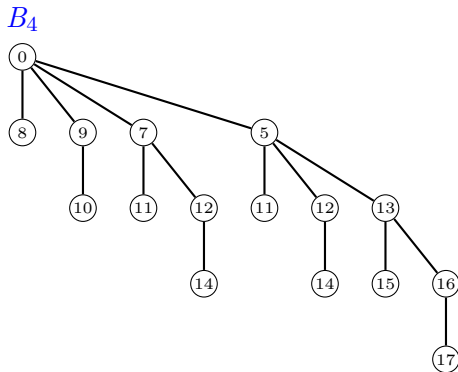
Binomial tree B_2



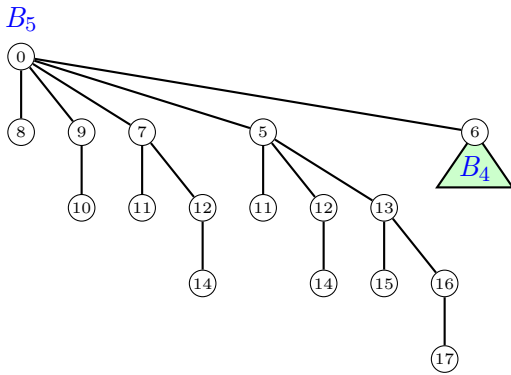
Binomial tree B_3



Binomial tree B_4



Binomial tree B_5



Binomial tree F0

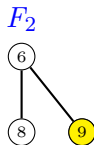
F_0

6

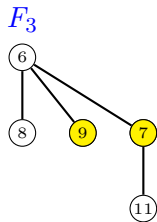
Binomial tree F1



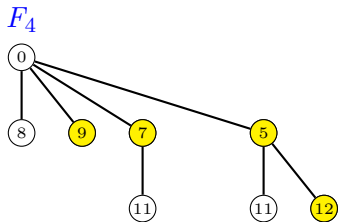
Binomial tree F_2



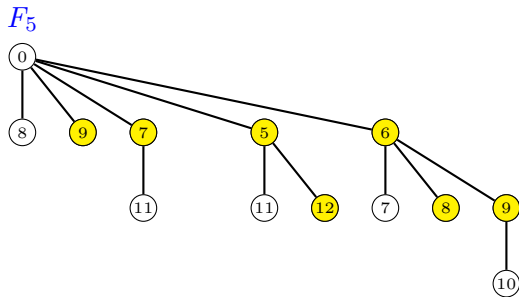
Binomial tree F_3



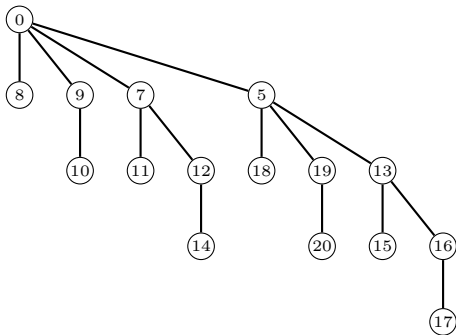
Binomial tree F_4



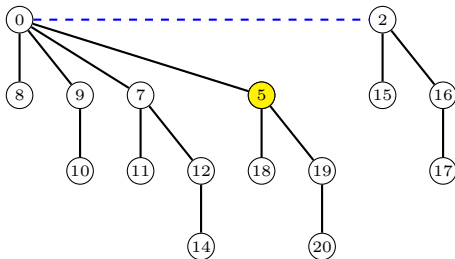
Binomial tree F_5



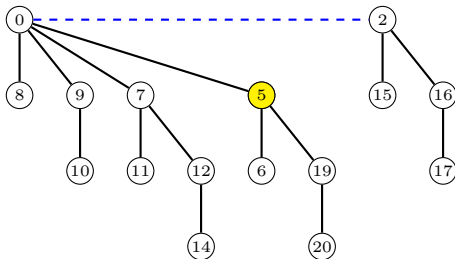
Fibonacci Heap Original



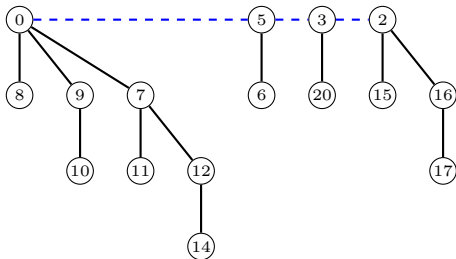
Fibonacci Heap: Decrease 13 to 2



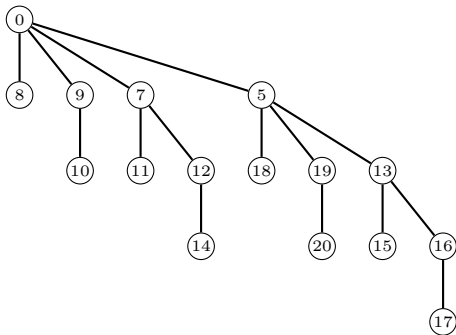
Fibonacci Heap: Decrease 18 to 6



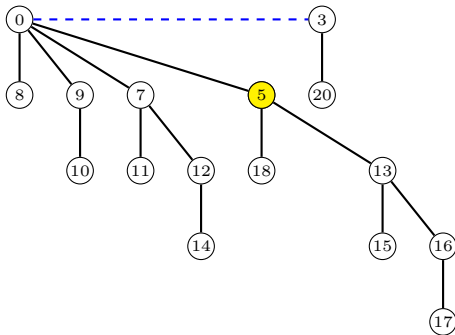
Fibonacci Heap: Decrease 19 to 3



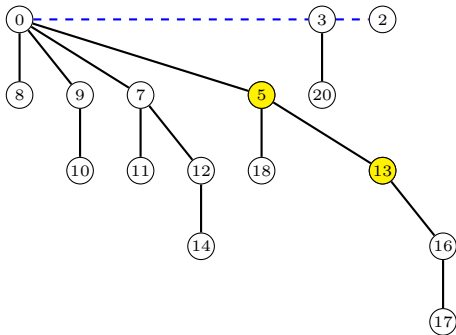
Fibonacci Heap Original



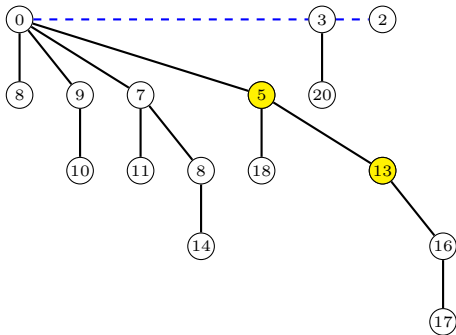
Fibonacci Heap Decrease 19 to 3



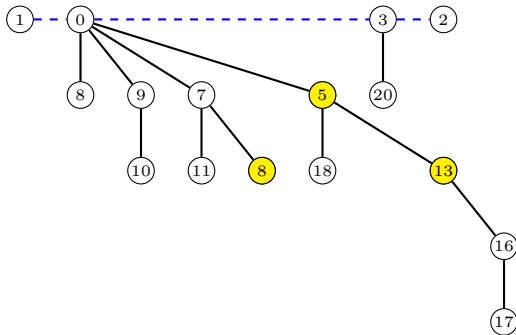
Fibonacci Heap Decrease 15 to 2



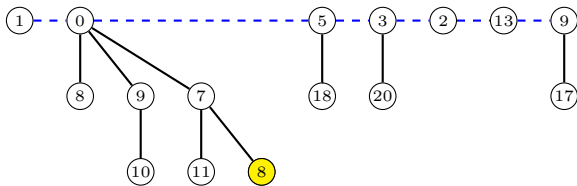
Fibonacci Heap Decrease 12 to 8



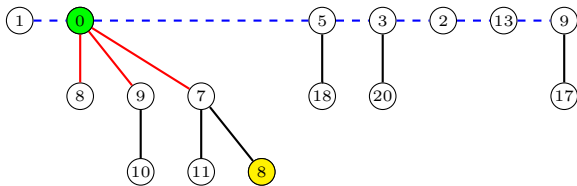
Fibonacci Heap Decrease 14 to 1



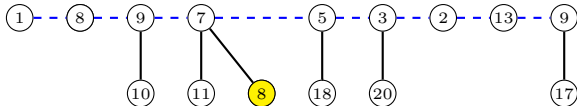
Fibonacci Heap Decrease 16 to 9



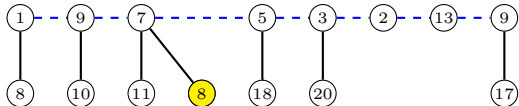
Fibonacci Heap ExtractMin



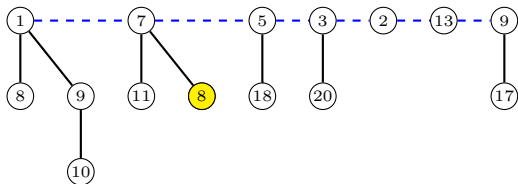
Fibonacci Heap ExtractMin2



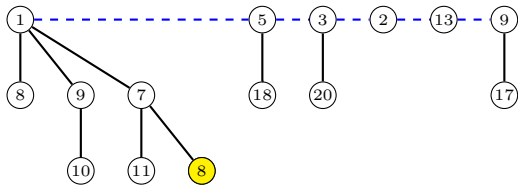
Fibonacci Heap Consolidate 1



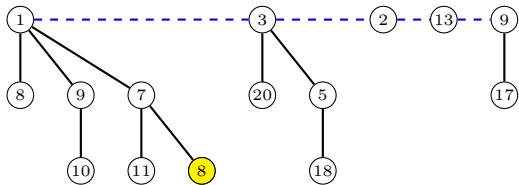
Fibonacci Heap Consolidate 2



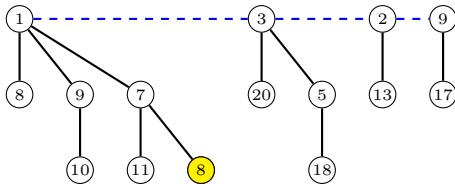
Fibonacci Heap Consolidate 3



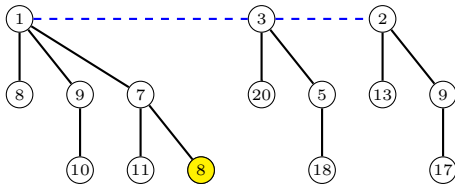
Fibonacci Heap Consolidate 4



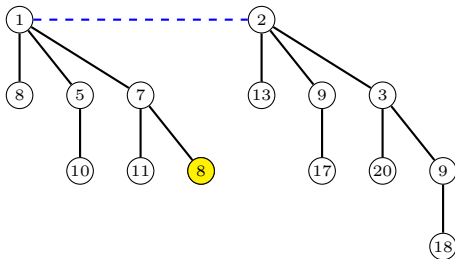
Fibonacci Heap Consolidate 5



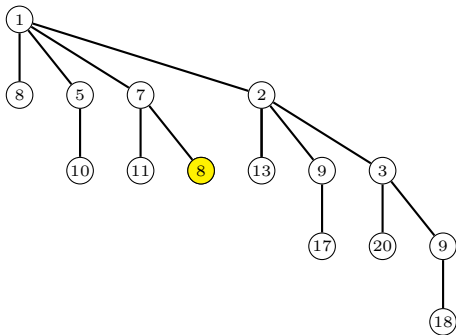
Fibonacci Heap Consolidate 6



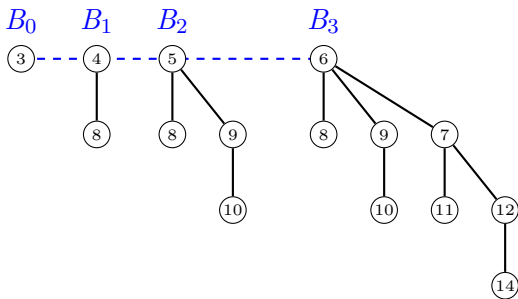
Fibonacci Heap Consolidate 7



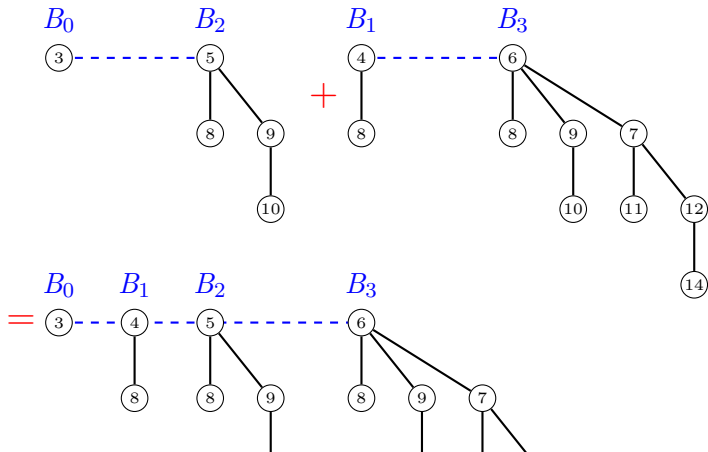
Fibonacci Heap Consolidate 8



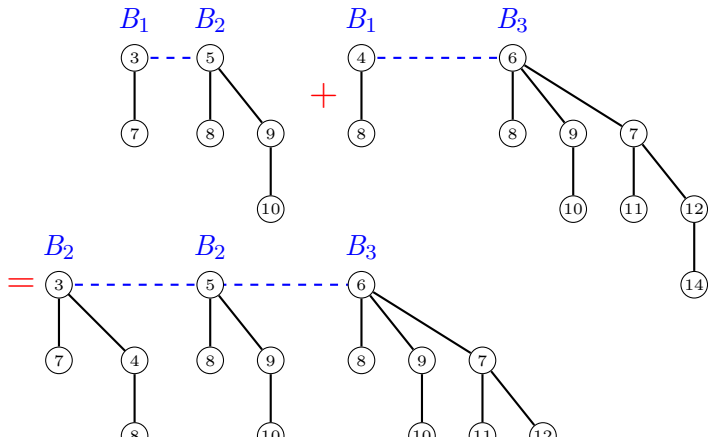
Binomial tree B_{0123} Union 1 Results



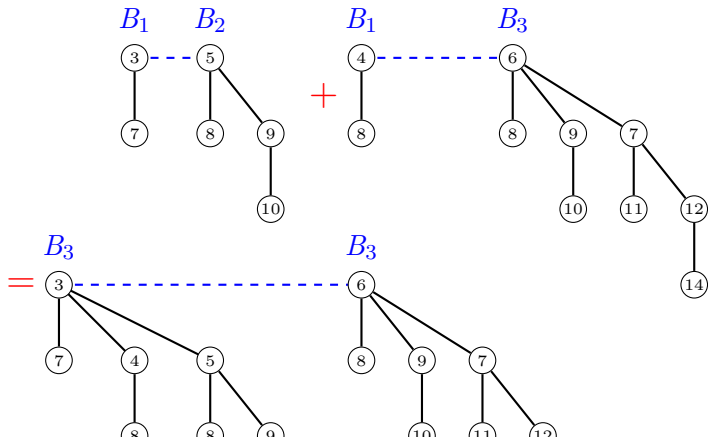
Binomial tree B_{0123} Union 1



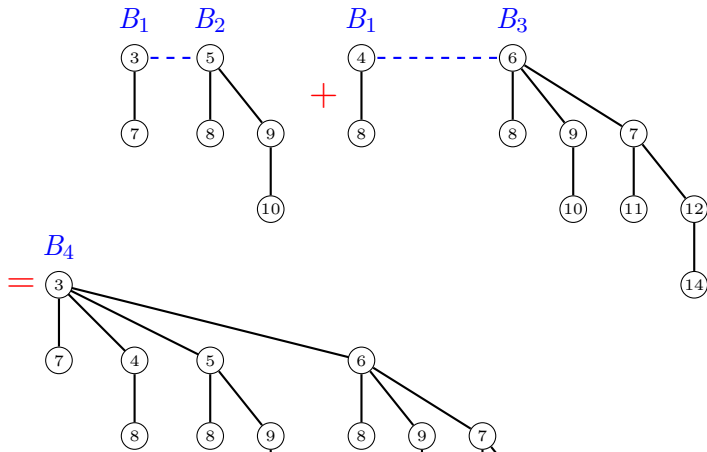
Binomial tree B_{0123} Union 2



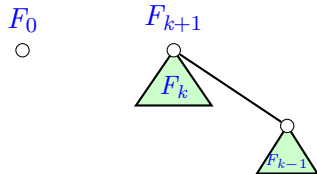
Binomial tree B_{0123} Union 2



Binomial tree B_{0123} Union 2



Fibonacci tree F_k



L8-LP example 3D step1

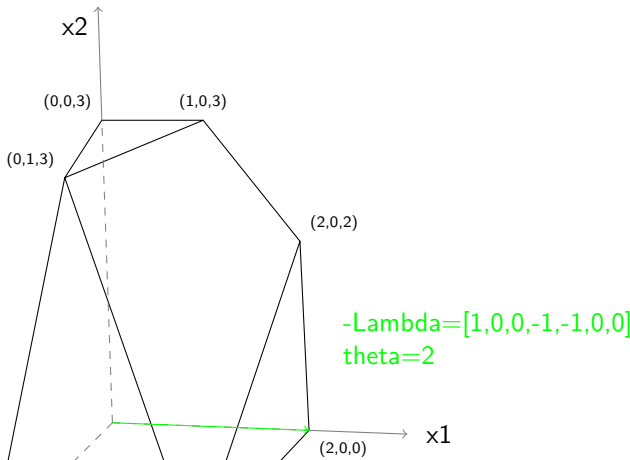


CHART 1

	—	$S1$	$S2$	$S3$	$S4$	$S5$...
—	0	0	0	0	0	0	
$T1$	0	1	2	3	4	5	6
$T2$	0	2	3	4	5	6	7
$T3$	0	3	4	5	6	7	8
$T4$	0	4	5	6	7	8	9
$T5$	0	5	6	7	8	9	10

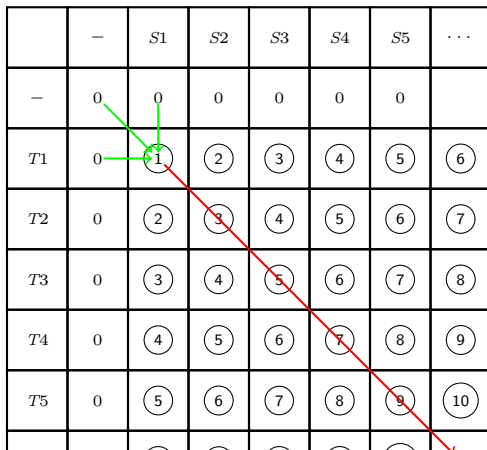
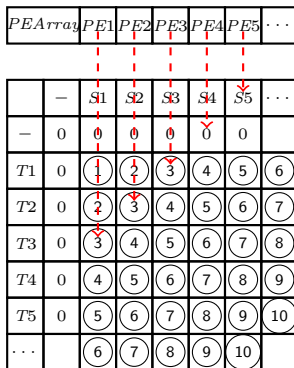


CHART 2



AAA

-0.3	0.1	0.4	0.6	0.4	0.6	0.4	0.6	0.4	0.6
------	-----	-----	-----	-----	-----	-----	-----	-----	-----

-0.3	0.1	0.4	0.7	0.9	1.1	1.3	1.5	1.7	1.9
	1.3	1.7	2.0	2.3	2.5	2.7	2.9	3.1	3.3
		0.9	1.3	1.6	1.9	2.1	2.3	2.5	2.7
			0.6	1.0	1.3	1.6	1.8	2.0	2.2
				0.4	0.8	1.2	1.4	1.7	1.9
					0.6	1.0	1.3	1.6	1.8
						0.4	0.8	1.2	1.4
							0.6	1.0	1.3

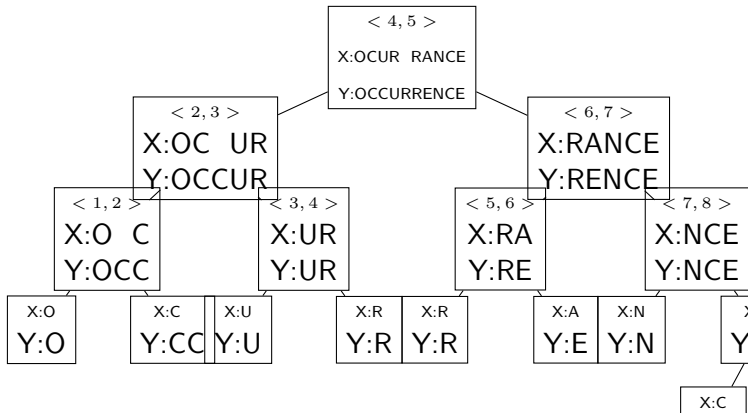
Hirschberg's algorithm: step 1

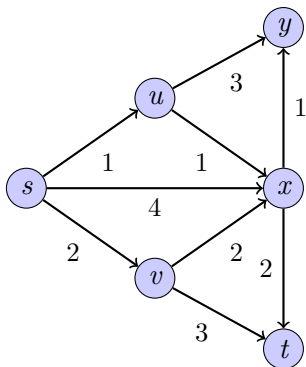
Y: ' ' O C U R												
X: ' ' O C C U R R E N C E	0	1	2	3	4	10	6	7	7	8	9	10
	1	0	1	2	3	8	5	6	6	7	8	9
	2	1	0	1	2	6	4	5	5	6	7	8
	3	2	1	1	2	5	3	4	4	5	6	7
	4	3	2	1	2	4	2	3	3	4	5	6
	5	4	3	2	1	2	1	2	2	3	4	5
	6	5	4	3	2	4	2	1	1	2	3	4
	7	6	5	4	3	5	2	1	0	1	2	3
	8	7	6	5	4	7	3	2	1	0	1	2
	9	8	7	6	5	9	4	3	2	1	0	1

Hirschberg's algorithm: step 2

Hirschberg's algorithm: step 3

Tree





	k=0	1	2	3	4	5
s	0	0	0	0	0	0
u	∞	1	1	1	1	1
v	∞	2	2	2	2	2
x	∞	4	2	2	2	2
y	∞	∞	4	3	3	3
t	∞	∞	5	4	4	4