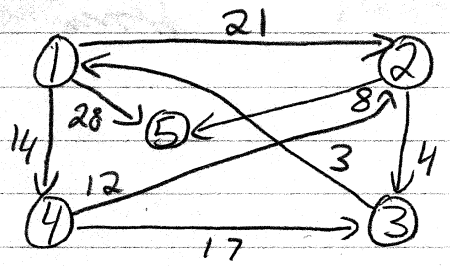
Analysis, Design of Algorithms

Jonathon Mann

1. Algorithm Adjust Min (a, i, n)

```
{
  j := 2i
  item = a[i]
  While (j < n)
  {
    if (j<n) and (a[j] > a[j+1]) then j := j+1
    if (item ≤ a[j]) then break
    a[⌊j/2⌋] := a[j]
    j : 2 j
  }
  a[⌊j/2⌋] := item
}
```

2.



Source = 1

$dist(2)_1 = cost(1,2) = 21$

$dist(3)_1 = cost(1,3) = \infty$

$dist(4)_1 = cost(1,4) = 14$

$dist(5)_1 = cost(1,5) = 28$

next vertex of least weight = 4

$dist(2)_2 = min\{dist(2)_1, dist(4)_1 + cost(4,2)\} = min\{21, 26\} = 21$

$dist(3)_2 = min\{dist(3)_1, dist(4)_1 + cost(4,3)\} = min\{\infty, 31\} = 31$

next vertex of least weight = 2

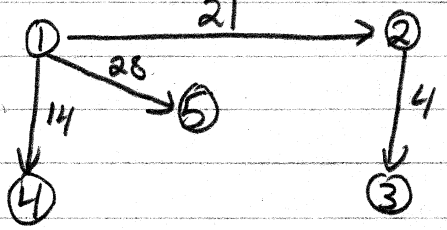
$dist(3)_3 = min\{dist(3)_2, dist(2)_2 + cost(2,3)\} = min\{31, 25\} = 25$

$dist(5)_2 = min\{dist(5)_1, dist(2)_2 + cost(2,5)\} = min\{28, 29\} = 28$

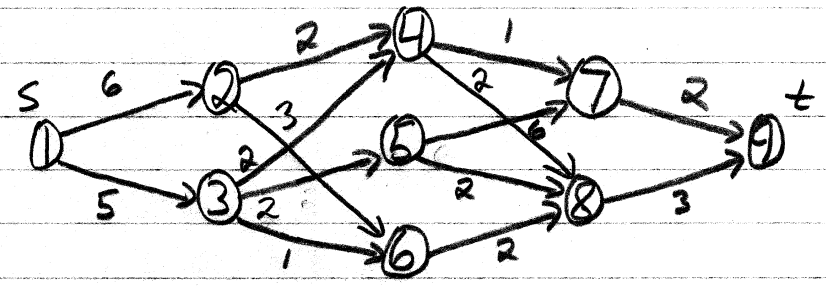
only edge from 3 is back to one, and no edges out of 5 exist

Final distances	Pathes
dist(1) = 0	Source
dist(2) = 21	1 -> 2
dist(3) = 25	1 -> 2 -> 3
dist(4) = 14	1 -> 4
dist(5) = 28	1 -> 5

Graph



3.



Vertex	1	2	3	4	5	6	7	8	9
cost	10	5	5	3	5	5	2	3	0
P	3	4	4	7	8	8	9	9	9

$Cost(9) = 0$

$cost(7) = c(7,9) = 2$

$Cost(8) = C(8,9) = 3$

$Cost(4) = \min\{c(4,7) + cost(7), c(4,8) + cost(8)\} = \min\{1+2, 2+3\} = 3$

$Cost(5) = \min\{c(5,7) + cost(7), c(5,8) + cost(8)\} = \min\{6+2, 2+3\} = 5$

$cost(6) = c(6,8) + cost(8) = 2 + 3 = 5$

$Cost(2) = \min\{c(2,4) + cost(4), c(2,6) + cost(6)\} = \min\{2+3, 3+5\} = 5$

$Cost(3) = \min\{c(3,4) + cost(4), c(3,5) + cost(5), c(3,6) + cost(6)\} = \min\{2+3, 2+5, 1+5\} = 5$

$Cost(1) = \min\{c(1,2) + cost(2), c(1,3) + cost(3)\} = \min\{6+5, 5+5\} = 10$

Path : 1 -> 3 -> 4 -> 7 -> 9