CS3383 Unit 2: Greedy

David Bremner

January 31, 2018

Contents

Huffman Coding

- DPV 5.2
- http://jeffe.cs.illinois.edu/teaching/algorithms/notes/07-greedy.pdf
- Huffman Coding is covered in §7.4

Prefix codes

Char	Freq	Symbol
A	70	0
B	3	001
C	20	01
D	37	11

- variable length symbols

Prefix codes

Char	Freq	Symbol
A	70	0
B	3	001
C	20	01
D	37	11

- variable length symbols
- avoiding ambiguous bitstreams: what is 001?

Prefix codes

Char	Freq	Symbol
A	70	0
B	3	001
C	20	01
D	37	11

- variable length symbols
- avoiding ambiguous bitstreams: what is 001?
- no symbol should be a prefix of another.

Prefix codes

Char	Freq	Symbol
A	70	0
B	3	001
C	20	01
D	37	11

- variable length symbols
- avoiding ambiguous bitstreams: what is 001?
- no symbol should be a prefix of another.
- if A is 0, what is D?

Huffman coding

Figure 5_10 in DPV

$$\text{cost}(T) = \sum_{i=1}^{n} f_i \text{depth}_i \qquad \text{(Avg cost)}$$

Lightest leaves are deepest

Figure 5_10b in DPV

- proof by swapping

Huffman Algorithm

```
Huffman(f[1..n])
H = priority queue of ind., by freq.
for i = 1 to n do
    H.insert(i)
for k = n+1 to 2n−1 do
    i = H.deletemin()
    j = H.deletemin()
    f[k] = f[i] + f[j]
    H.insert(k)
    children[k] = (i,j)
```

Correctness (Board)

Huffman yields an optimal prefix code (induction)

Contents

Minimum spanning tree

Definition (Minimum Spanning Tree)

Given $G = (V, E)$, $w : E \to \mathbb{R}$, a minimum spanning tree T is a spanning tree (i.e. connecting all vertices) that minimizes $\mathrm{cost}(T) = \sum_{e \in T} w(e)$

Minimum Spanning trees

Figure 5_0 in DPV

Is this solution unique?

Minimum Spanning trees

Figure 5_0 in DPV

Is this solution unique?

Figure 5_1 in DPV

How about this one?

Cut Property

Figure 5_2 in DPV

Lemma (Board)

Let T be a minimum spanning tree, $X \subset T$ s.t. X does not connect $(S, V - S)$. Let e be the lightest edge from S to $V - S$. $X \cup e$ is part of some MST.

Generic MST

$X \leftarrow \{\}$
while $|X| < |V| - 1$ **do**
    Choose S s.t. X does not connect $(S, V - S)$
    Add the lightest crossing edge to X
end while

Figure 5_8a in DPV

Greedy Algorithms in General

Discrete Optimization Problems

- solution defined by a sequence of choices
- solutions are ranked from best to worst

Greedy Algorithms in General

Discrete Optimization Problems

- solution defined by a sequence of choices
- solutions are ranked from best to worst

Greedy Design Strategy

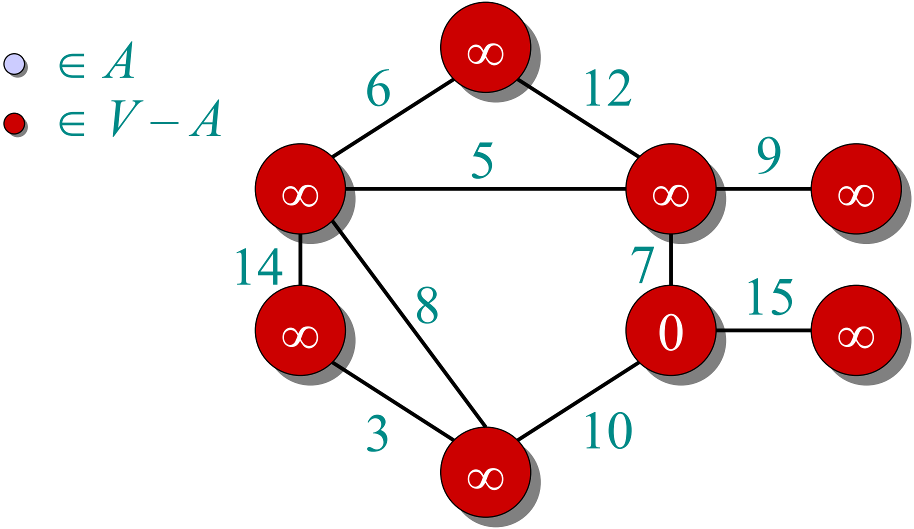
- Each choice leaves one smaller subproblem
- Prove that $\exists$ an optimal solution that makes the greedy choice
- Show that the greedy choice, combined with an optimal solution to the subproblem, yields an optimal solution to the original problem.

Prim's Algorithm

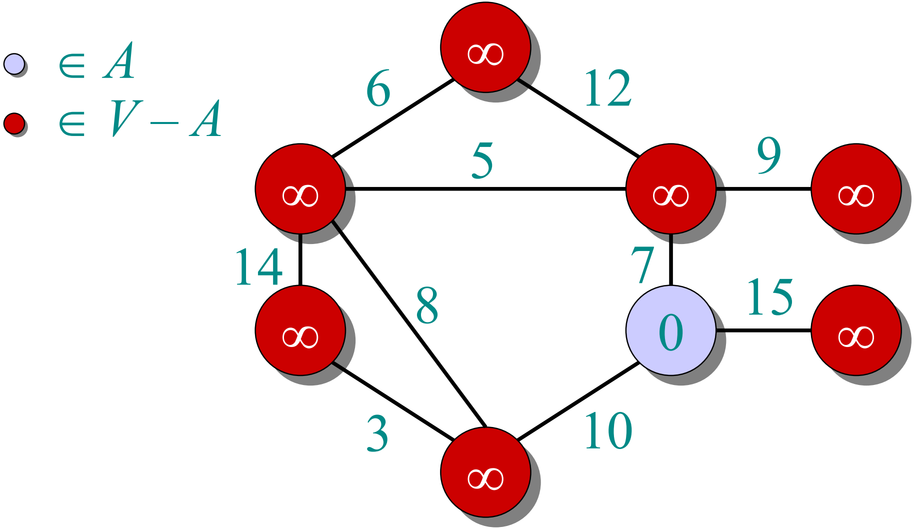
Figure 5_8 in DPV

S = nodes reached so far

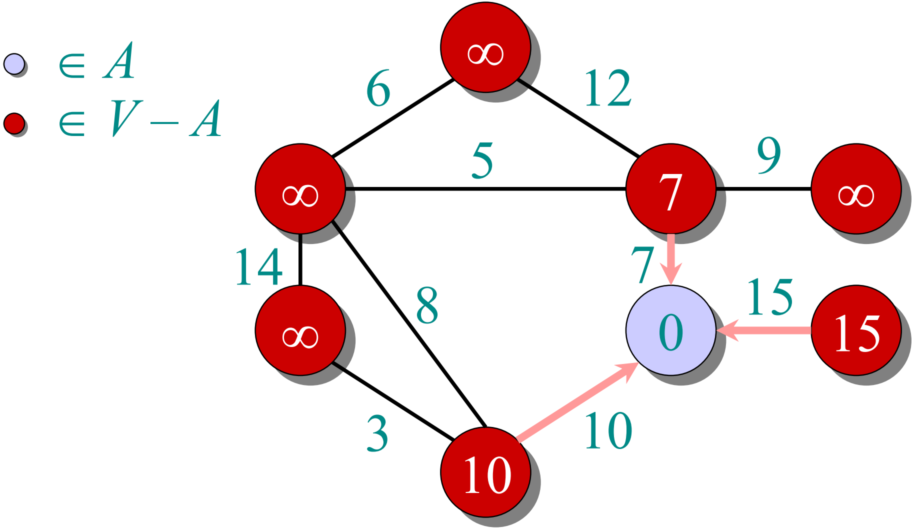
Example of Prim's algorithm



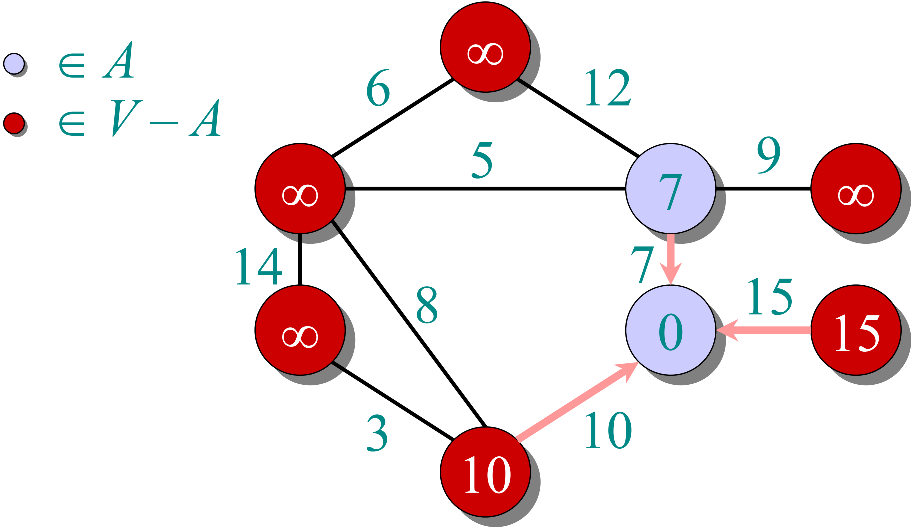
Example of Prim's algorithm



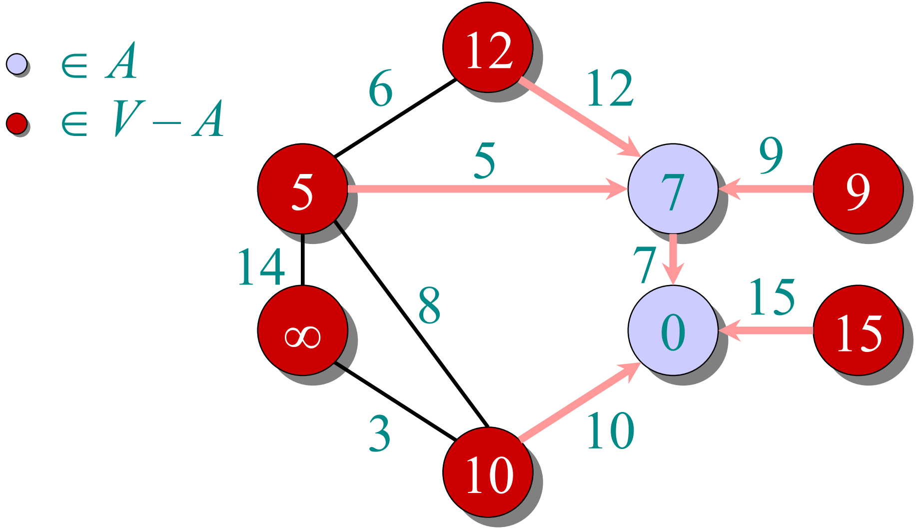
Example of Prim's algorithm



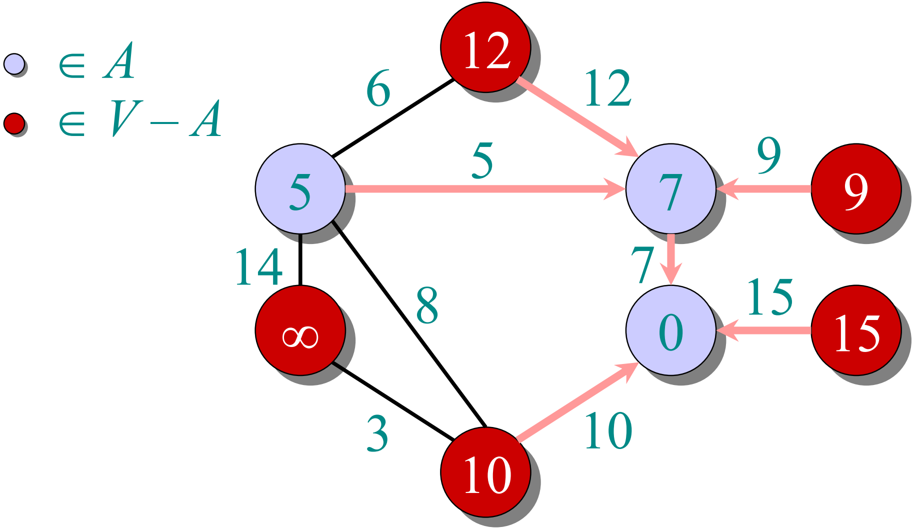
Example of Prim's algorithm



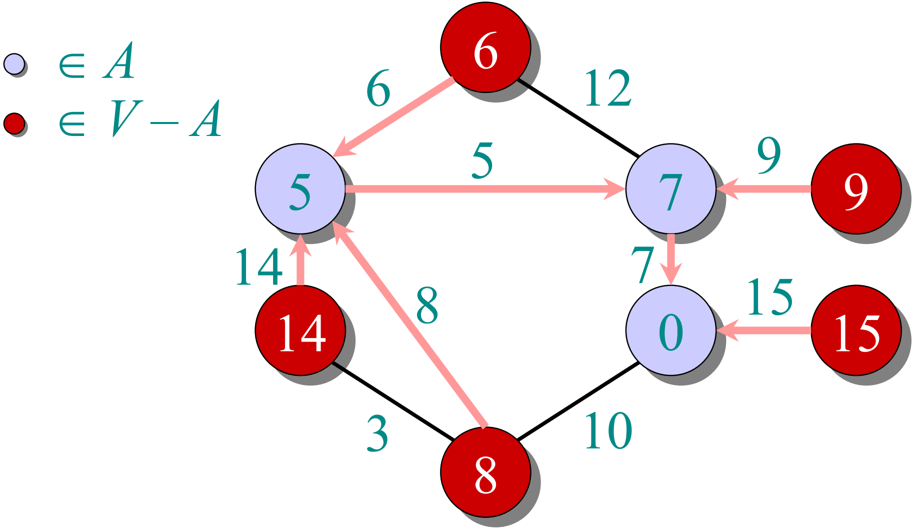
Example of Prim's algorithm



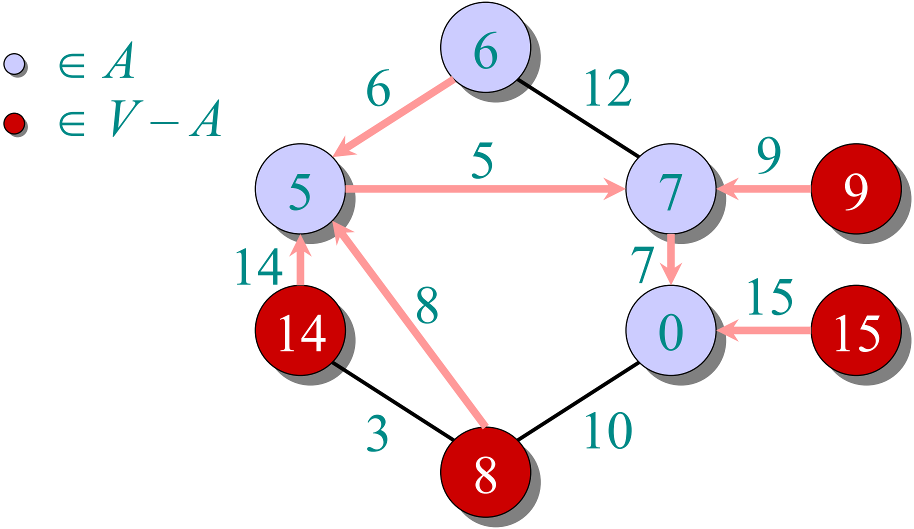
Example of Prim's algorithm



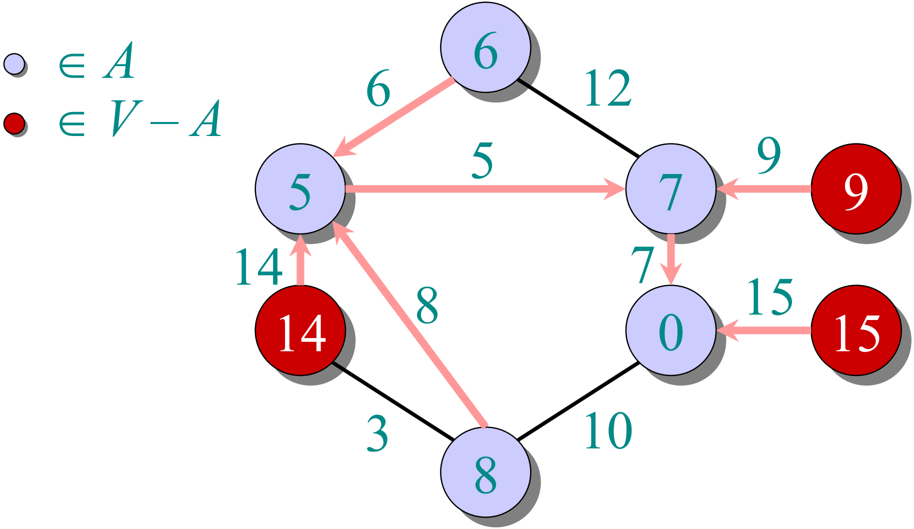
Example of Prim's algorithm



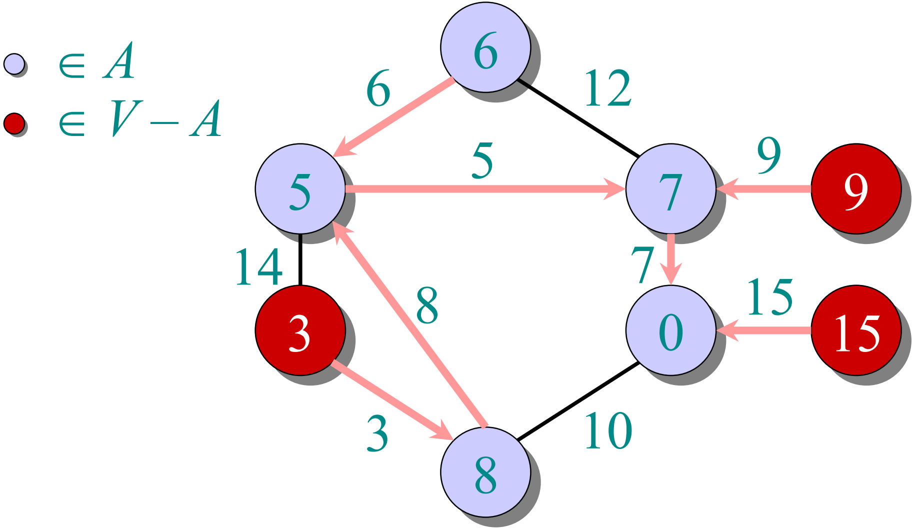
Example of Prim's algorithm



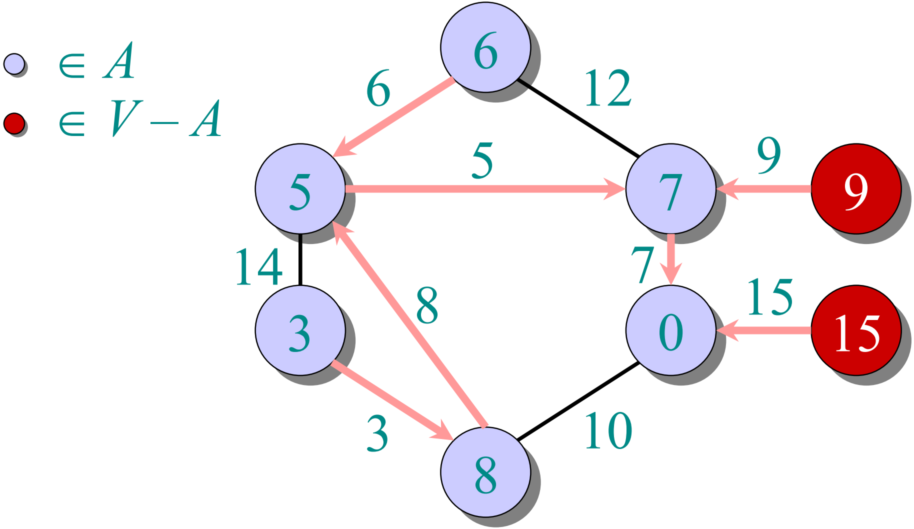
Example of Prim's algorithm



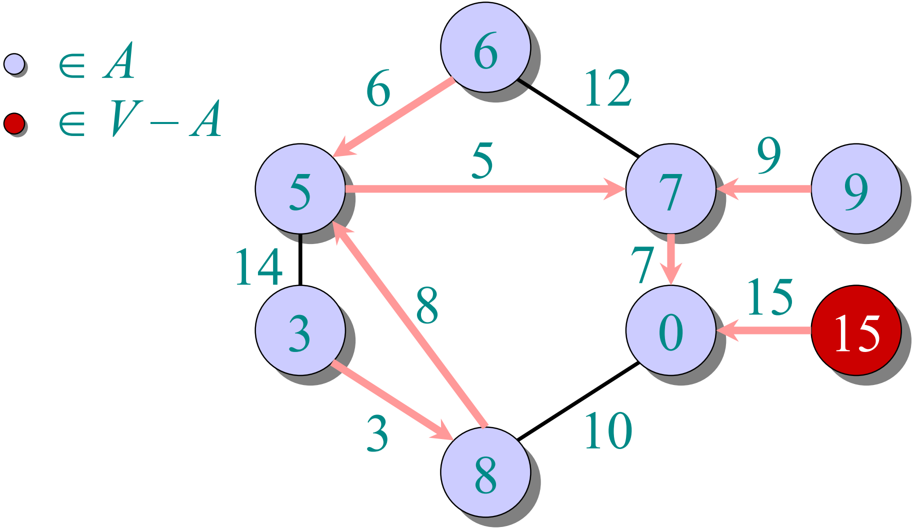
Example of Prim's algorithm



Example of Prim's algorithm

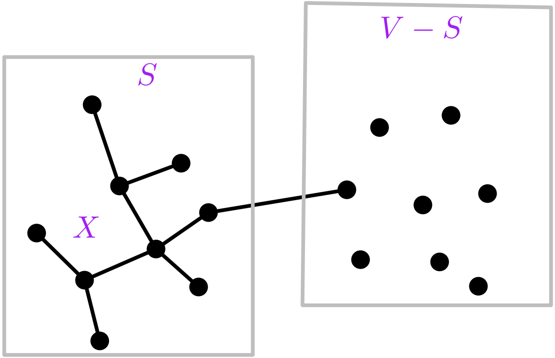


Example of Prim's algorithm



Prim's Algorithm

u_0 = arbitrary vertex
cost$(u_0) = 0$; cost$(v) = \infty$, $v \neq u_0$
for $v \in V$: enq(H, v)
while H is not empty **do**
    v = deletemin(H)
    for $e = \{v, z\}$, $e \in E$, $z \in H$ **do**
        if cost$(z) > w(v, z)$ **then**
            cost$(z) = w(v, z)$
            prev$(z) = v$
            decreasekey(H,z)
        end if
    end for
end while



Analysis of Prim's Algorithm (board)

- Correctness follows from the cut property, induction
- Closely connected with the Djikstra's Shortest path algorithm; only two lines change
- Tree can be read back from prev
- Cost is dominated by priority queue operations