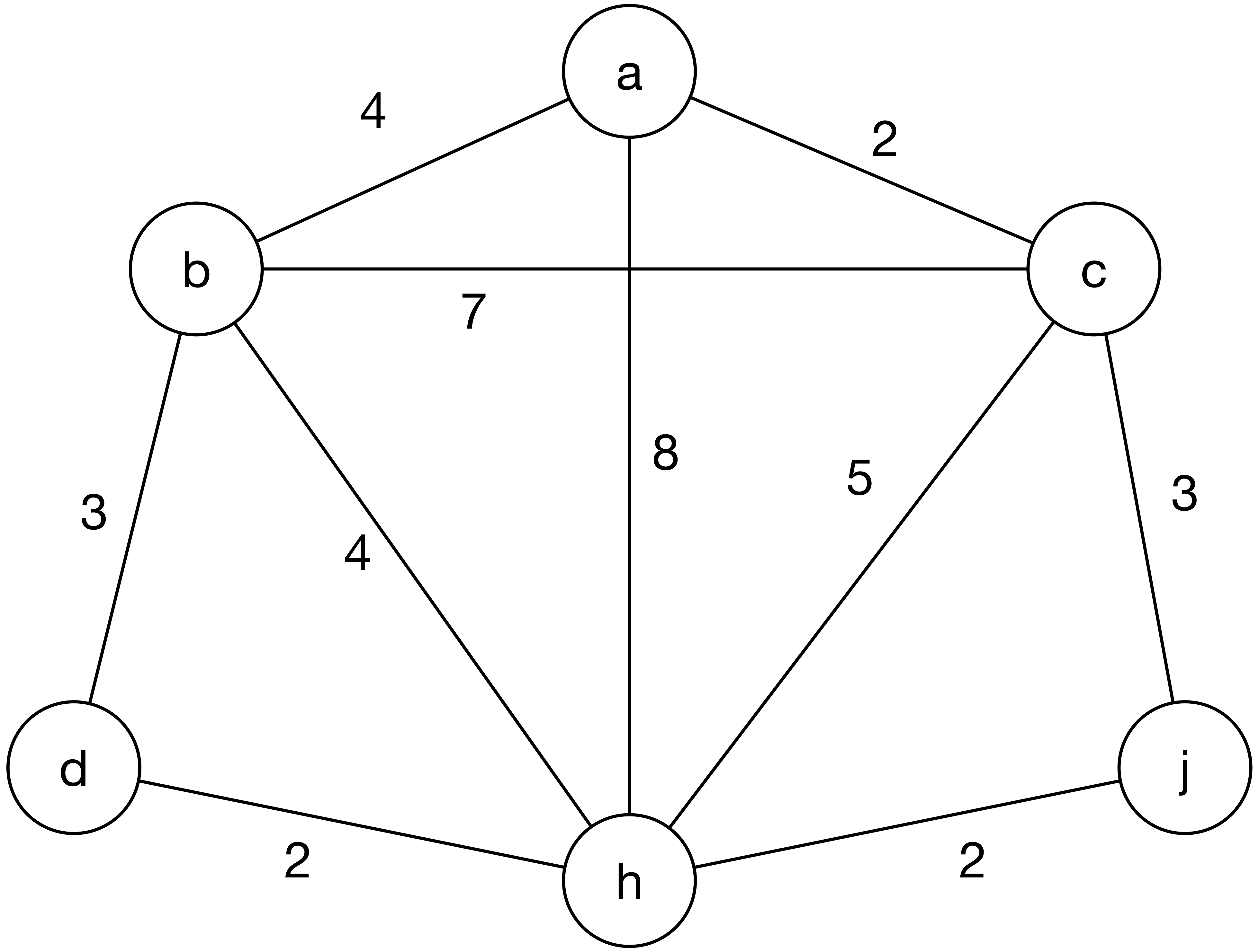
THE UNIVERSITY OF VERMONT
COLLEGE OF ENGINEERING &
MATHEMATICAL SCIENCES

Minimum Spanning Tree

Kruskal's algorithm

CS 124 / Department of Computer Science

Minimum spanning tree



$$G = (V, E)$$

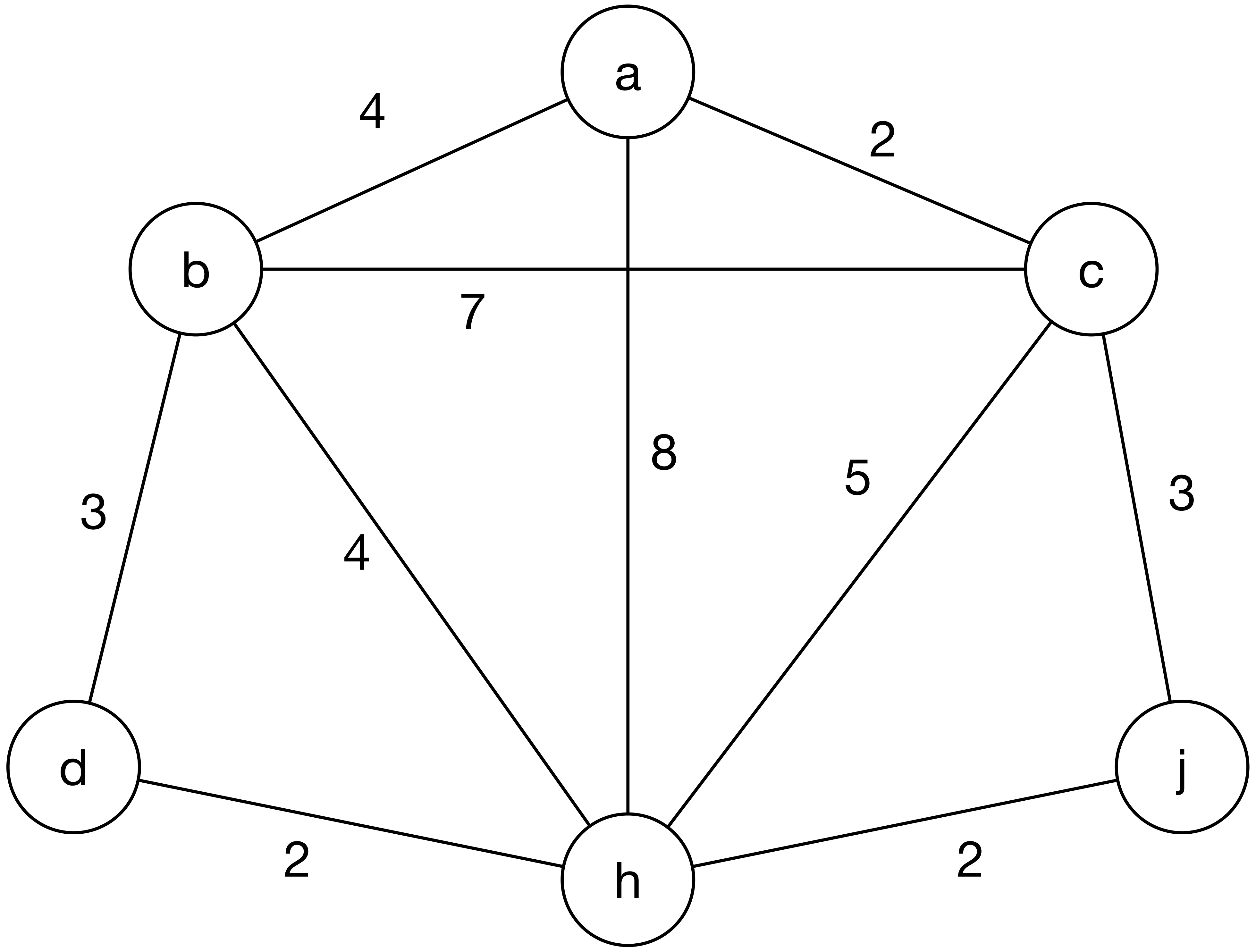
$$G' = (V, E') \text{ with } E' \subseteq E$$

$$G' \text{ is a tree}$$

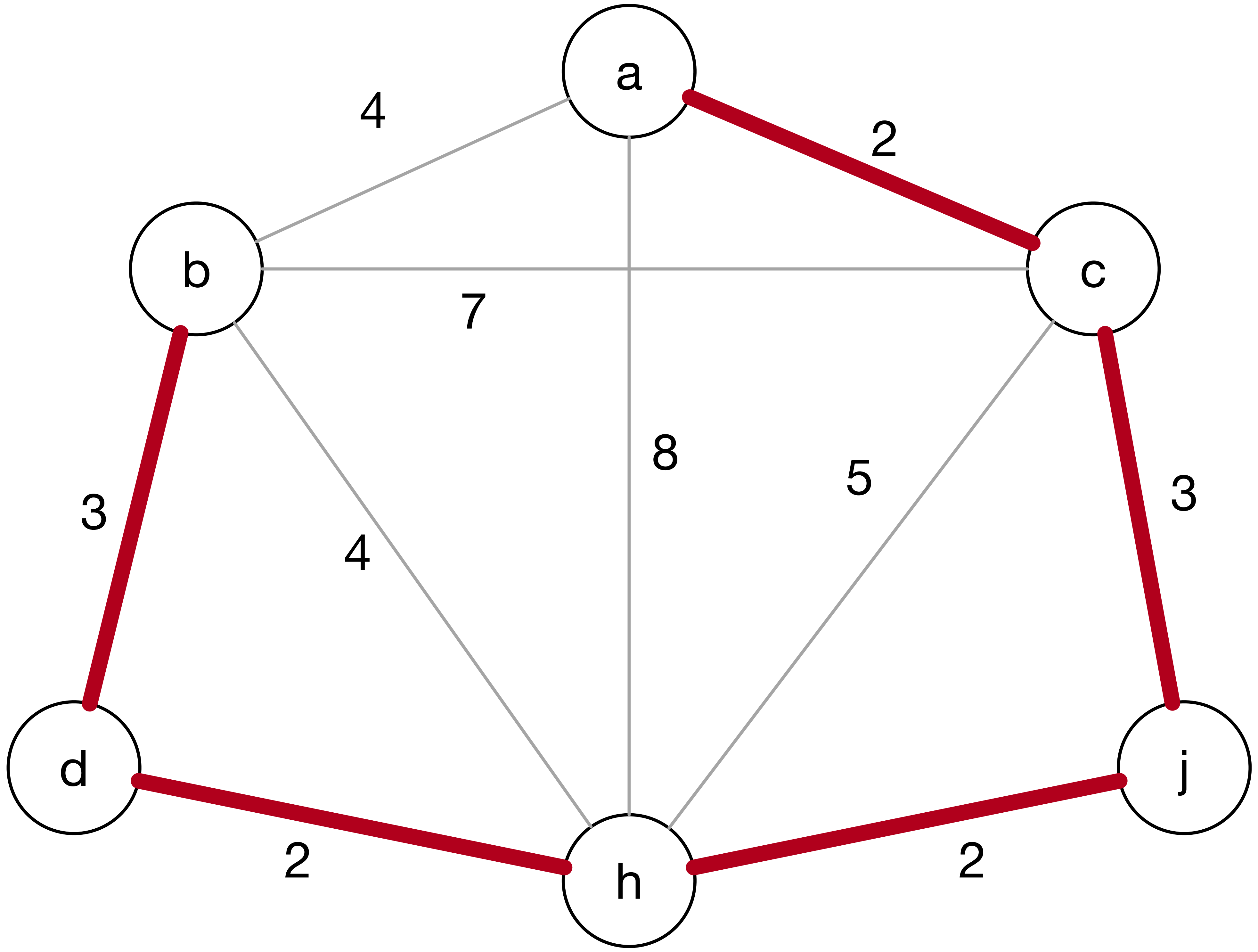
$$\operatorname{argmin} \sum_{(u,v) \in E'} w_{u,v}$$

Minimum spanning tree

$$\sum_{(u,v)\in E'} w_{u,v} = 12$$



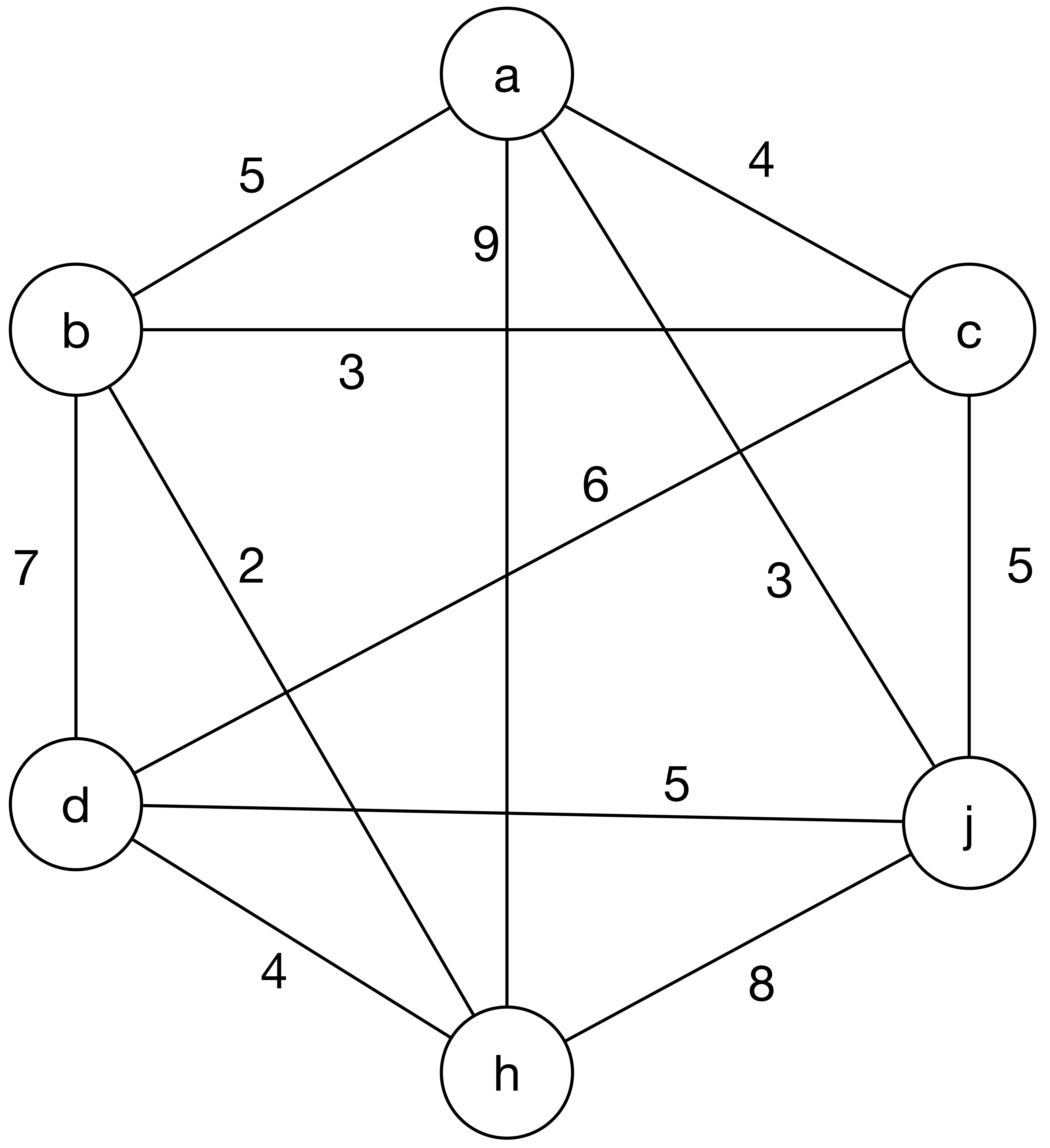
$$G = (V, E)$$



$$G' = (V, E') \text{ with } E' \subseteq E$$

Minimum spanning tree

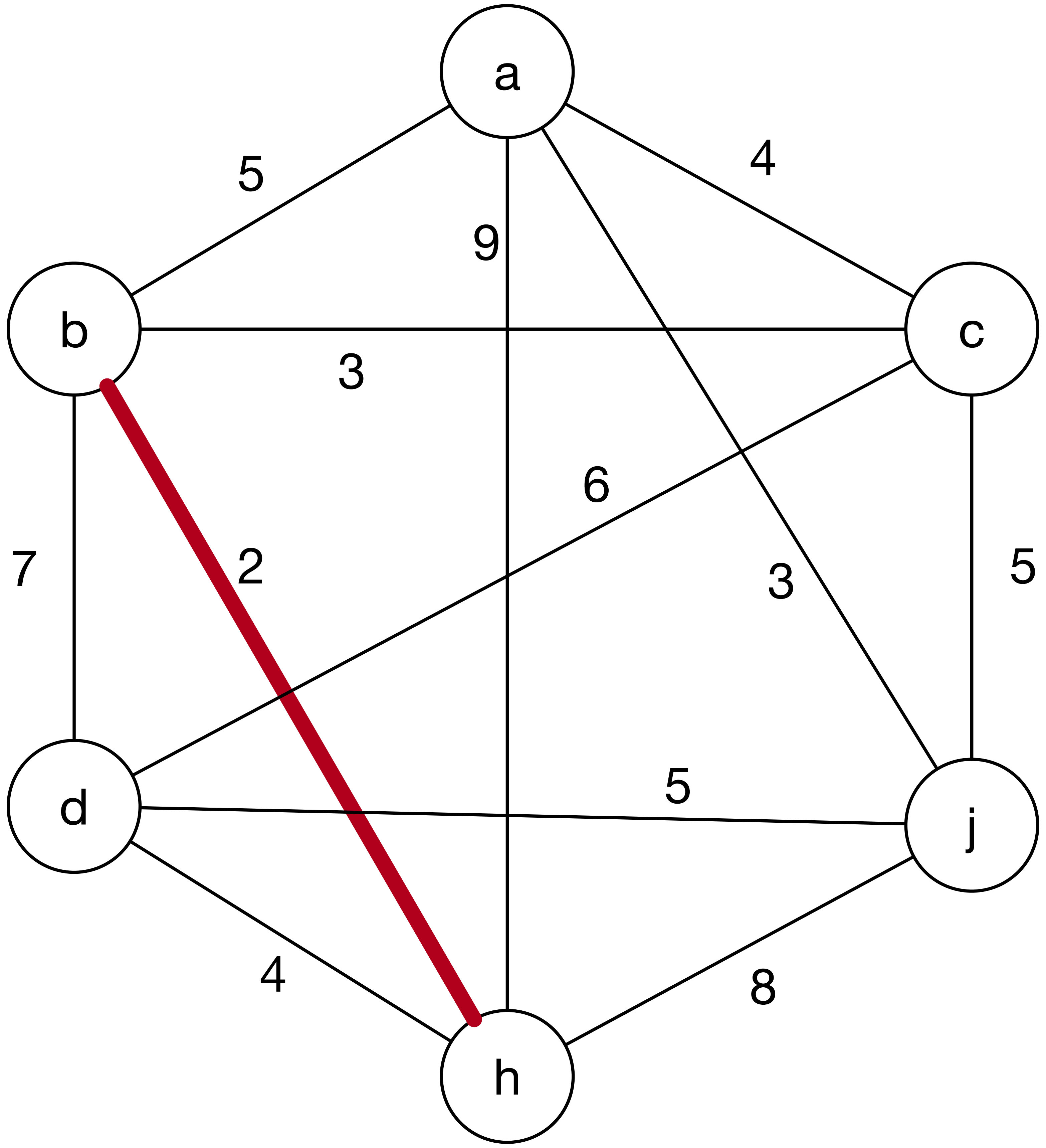
Kruskal's algorithm



edge	weight
bh	2
aj	3
bc	3
ac	4
dh	4
ab	5
cj	5
dj	5
cd	6
bd	7
jh	8
ah	9

Minimum spanning tree

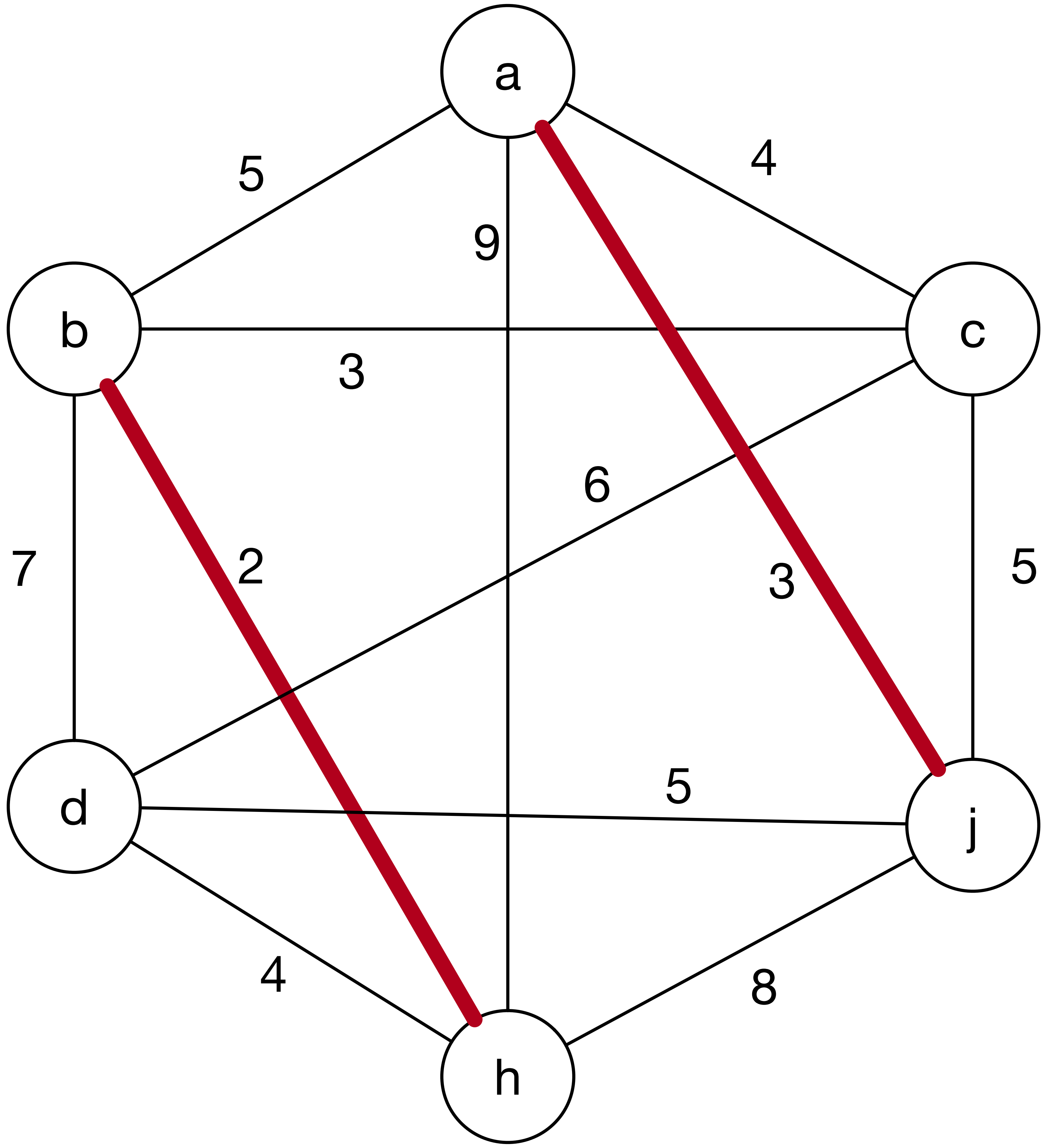
Kruskal's algorithm



edge	weight
bh	2
aj	3
bc	3
ac	4
dh	4
ab	5
cj	5
dj	5
cd	6
bd	7
jh	8
ah	9

Minimum spanning tree

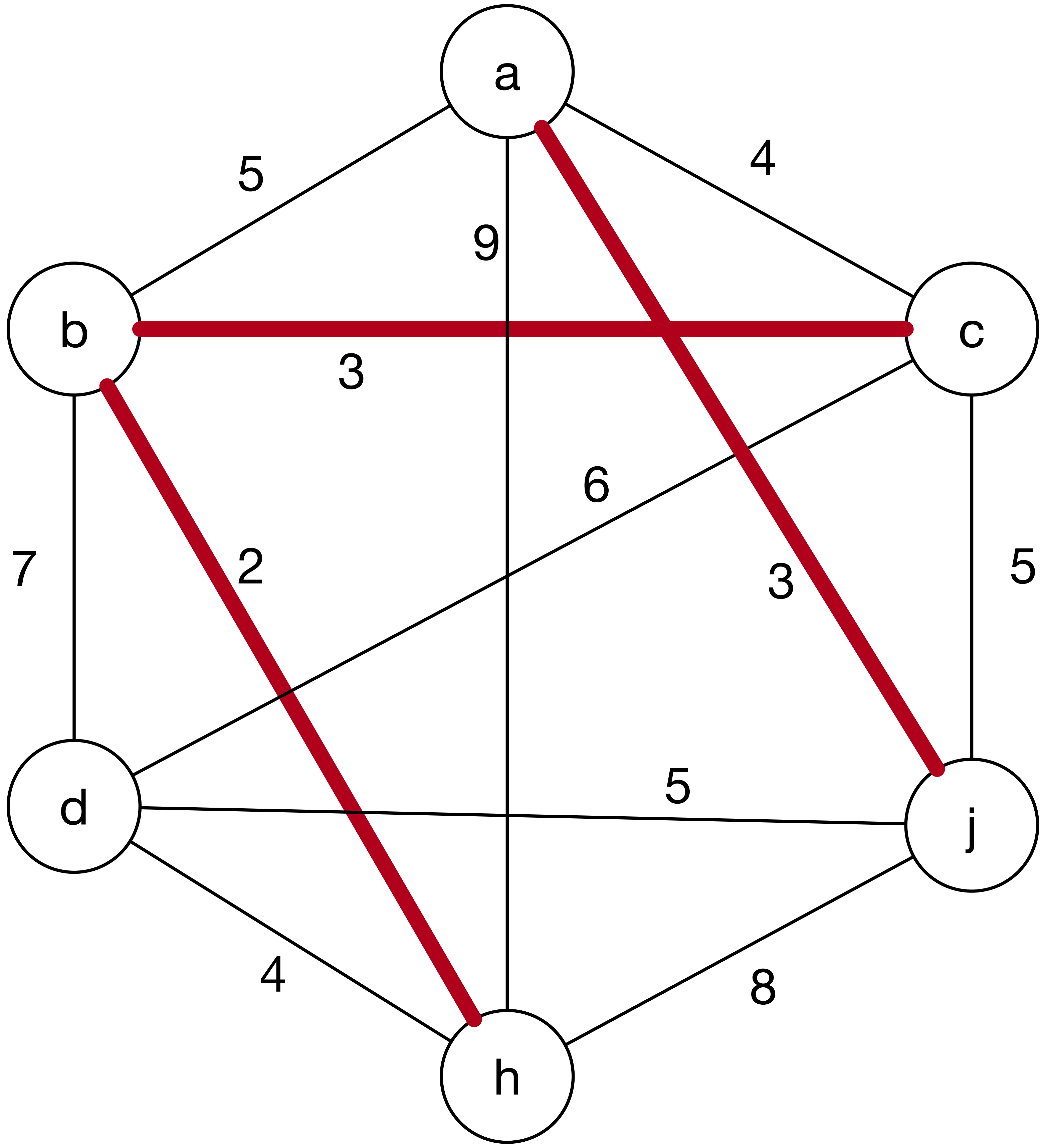
Kruskal's algorithm



edge	weight
bh	2
aj	3
bc	3
ac	4
dh	4
ab	5
cj	5
dj	5
cd	6
bd	7
jh	8
ah	9

Minimum spanning tree

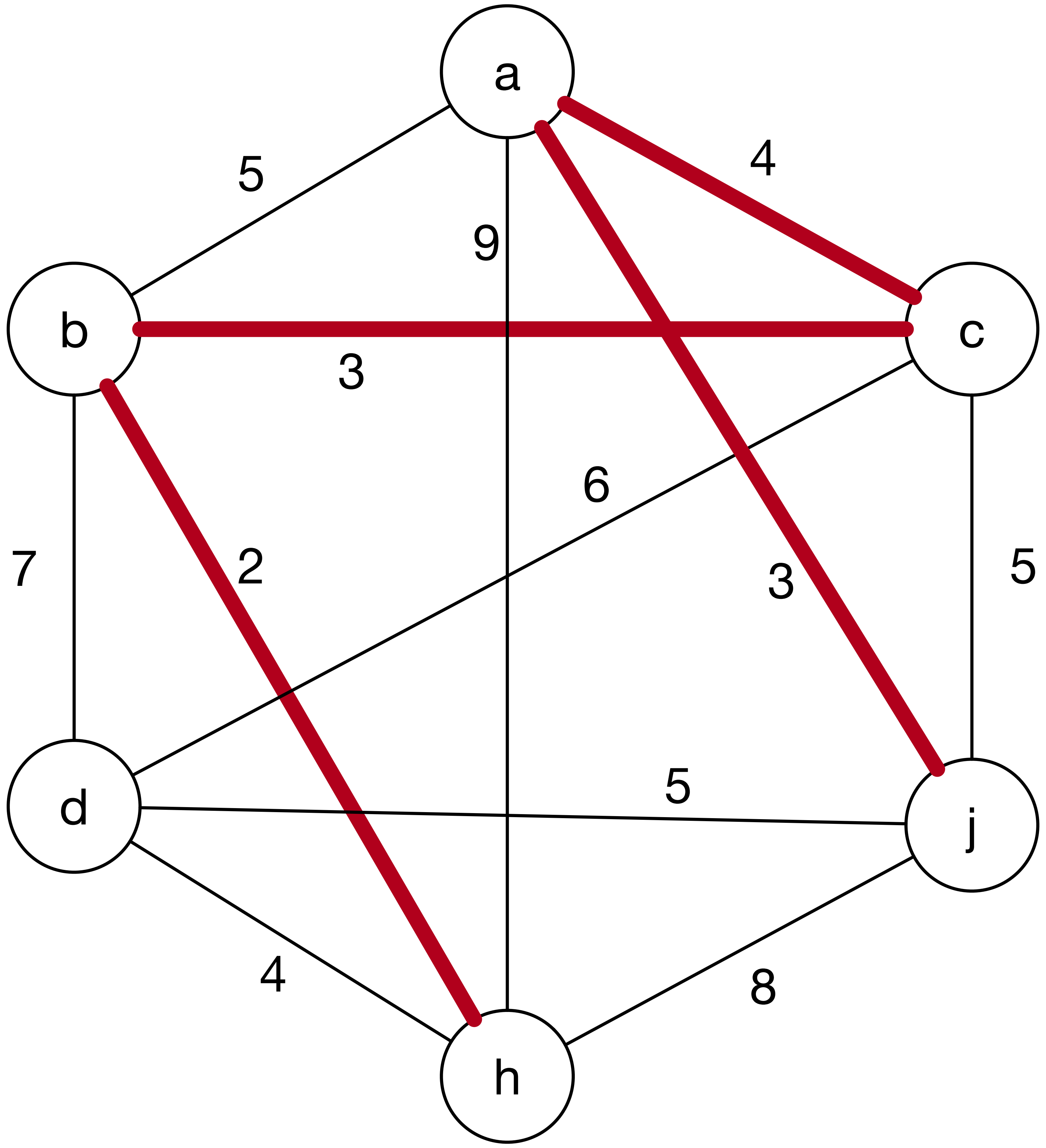
Kruskal's algorithm



edge	weight
bh	2
aj	3
bc	3
ac	4
dh	4
ab	5
cj	5
dj	5
cd	6
bd	7
jh	8
ah	9

Minimum spanning tree

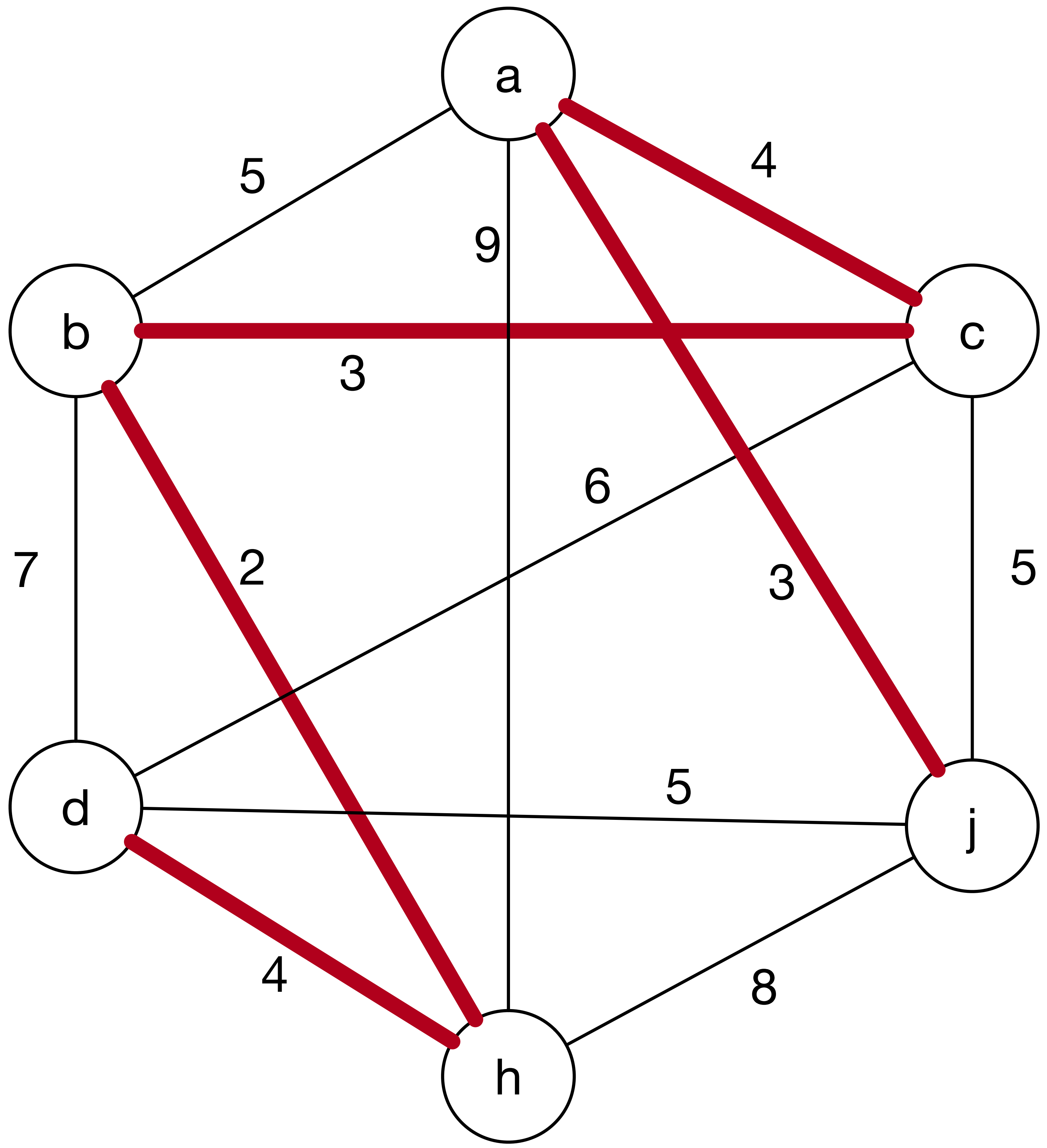
Kruskal's algorithm



edge	weight
bh	2
aj	3
bc	3
ac	4
dh	4
ab	5
cj	5
dj	5
cd	6
bd	7
jh	8
ah	9

Minimum spanning tree

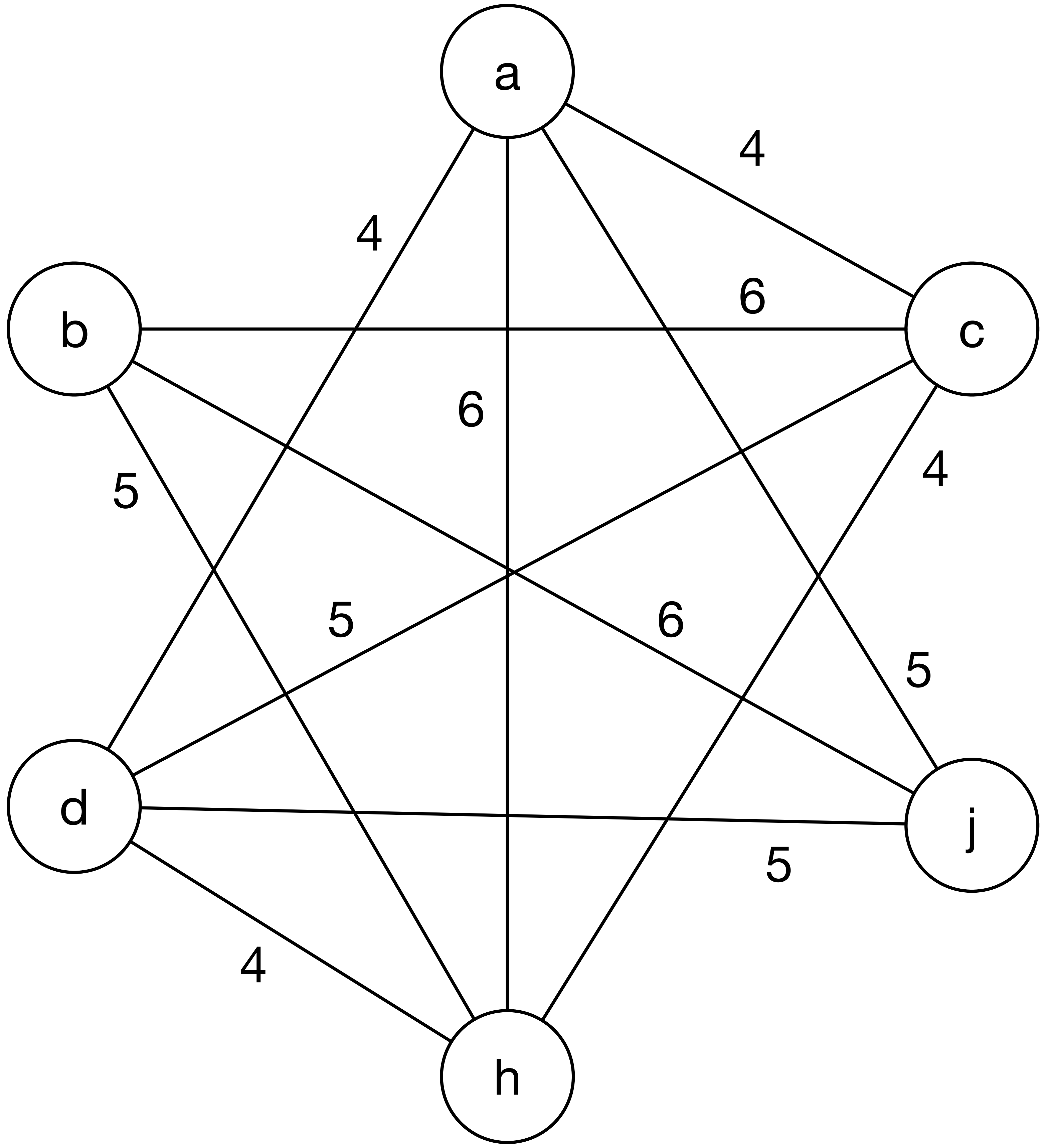
Kruskal's algorithm



edge	weight
bh	2
aj	3
bc	3
ac	4
dh	4
ab	5
cj	5
dj	5
cd	6
bd	7
jh	8
ah	9

Minimum spanning tree

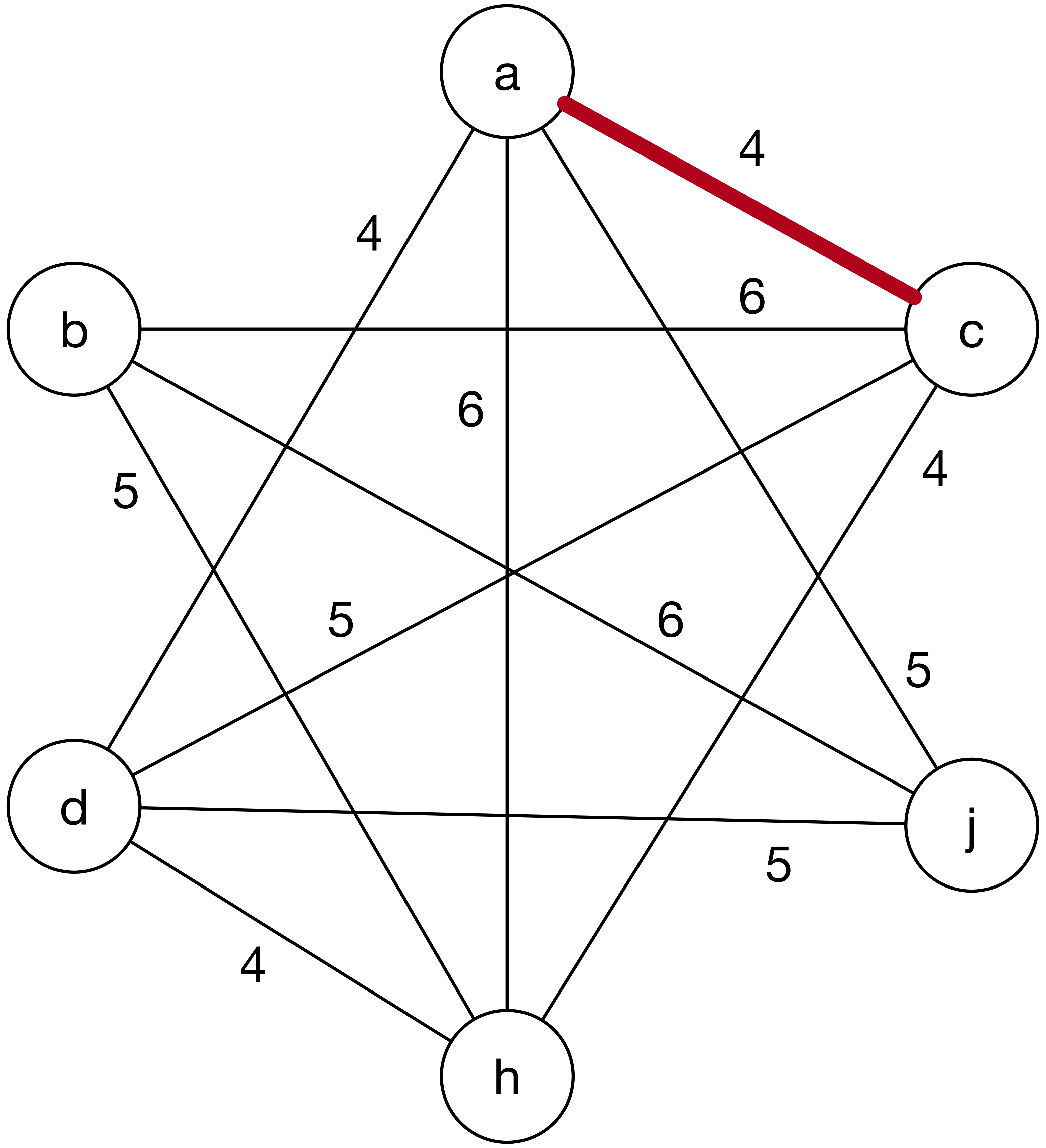
Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
dh	4
aj	5
dj	5
cd	5
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

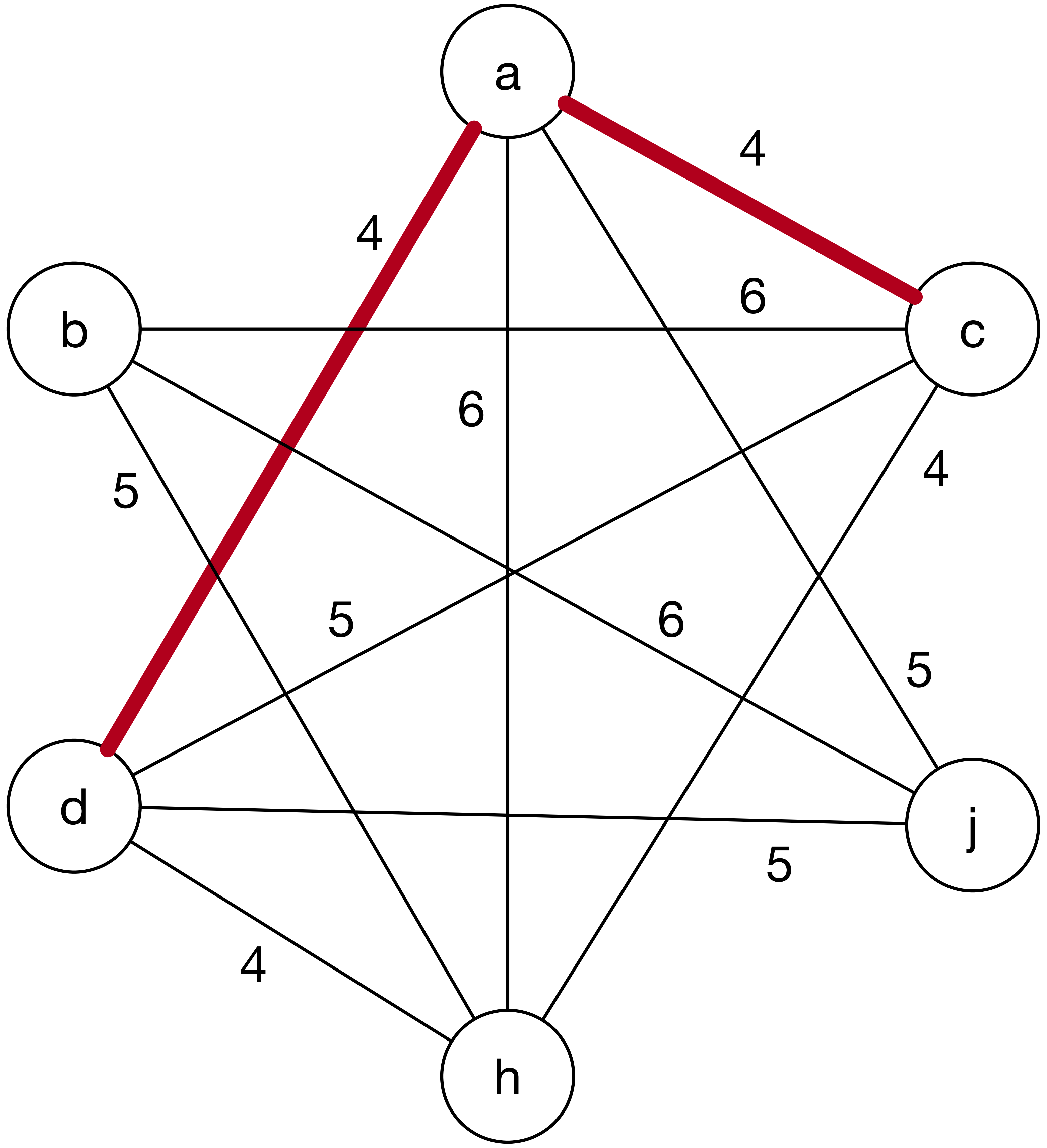
Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
dh	4
aj	5
dj	5
cd	5
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

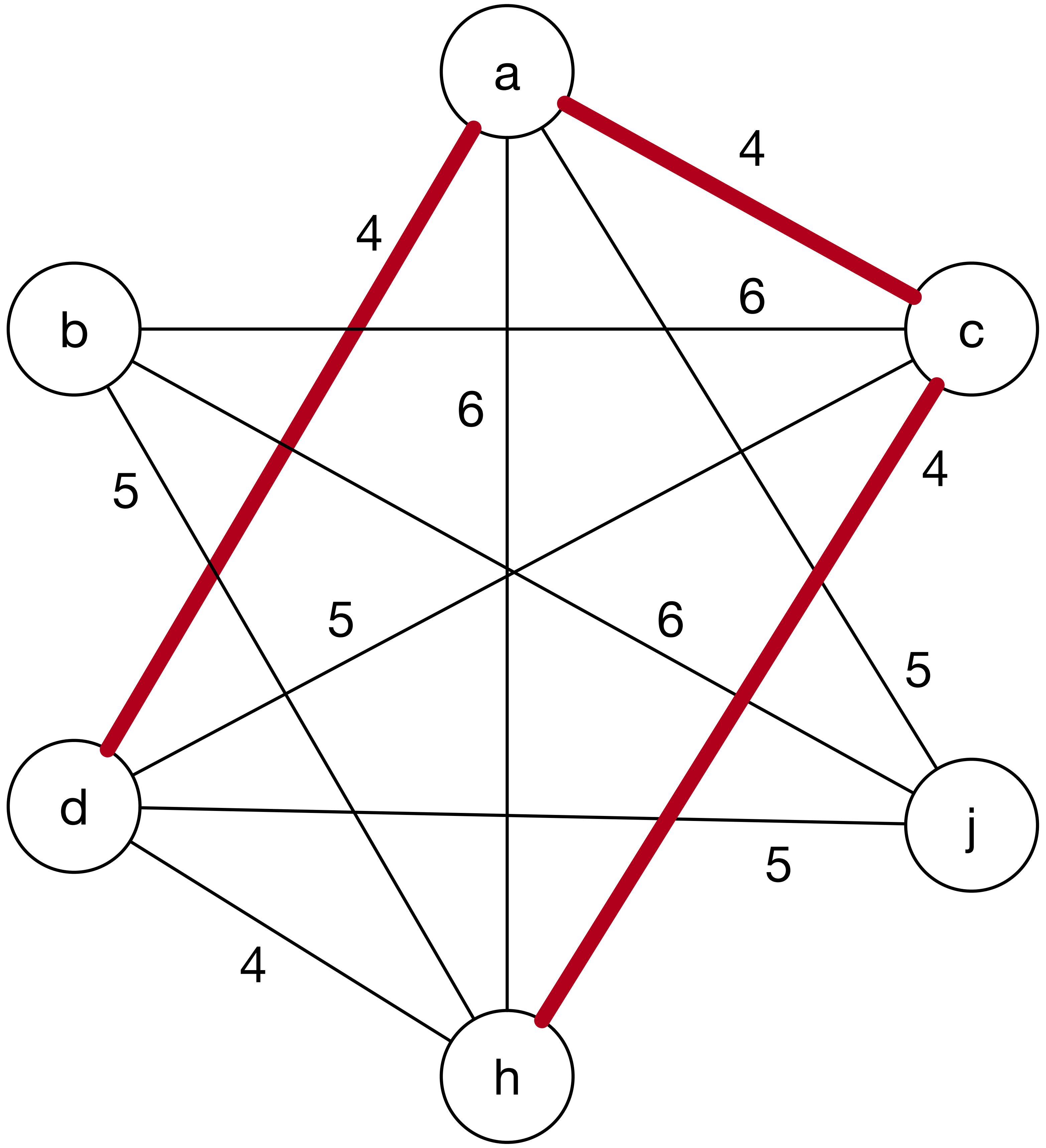
Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
dh	4
aj	5
dj	5
cd	5
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

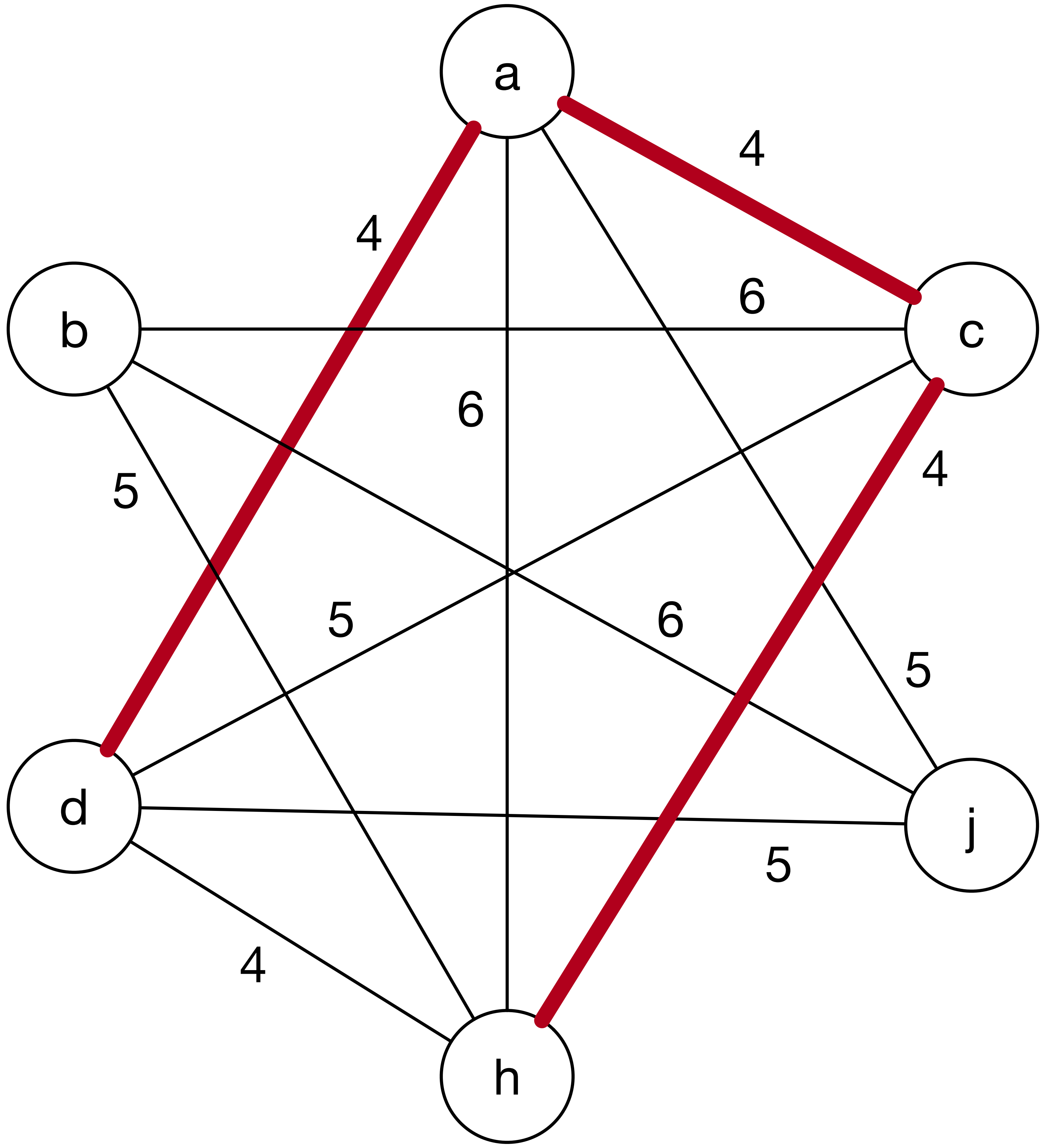
Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
dh	4
aj	5
dj	5
cd	5
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

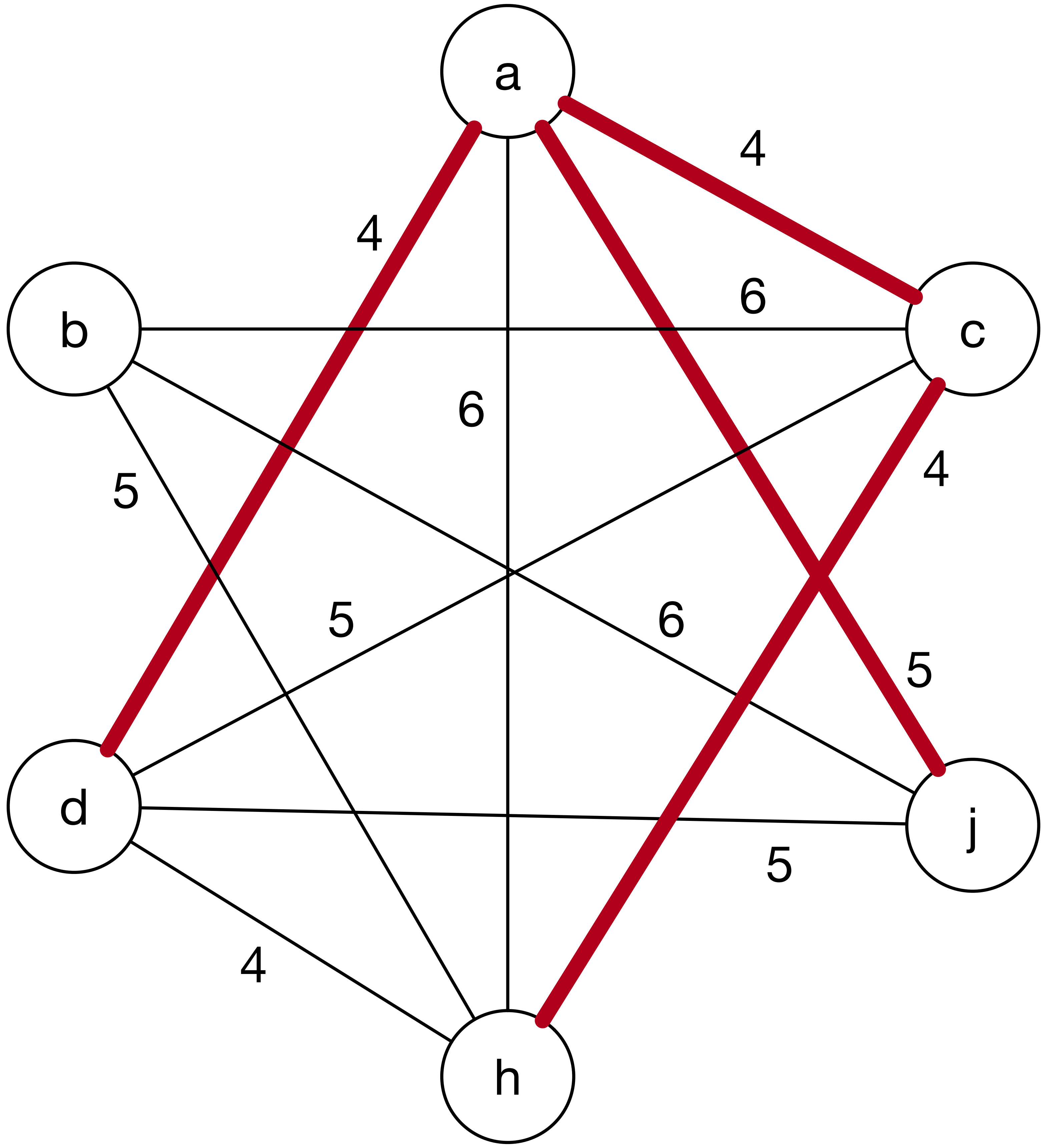
Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
~~dh~~	~~4~~
aj	5
dj	5
cd	5
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

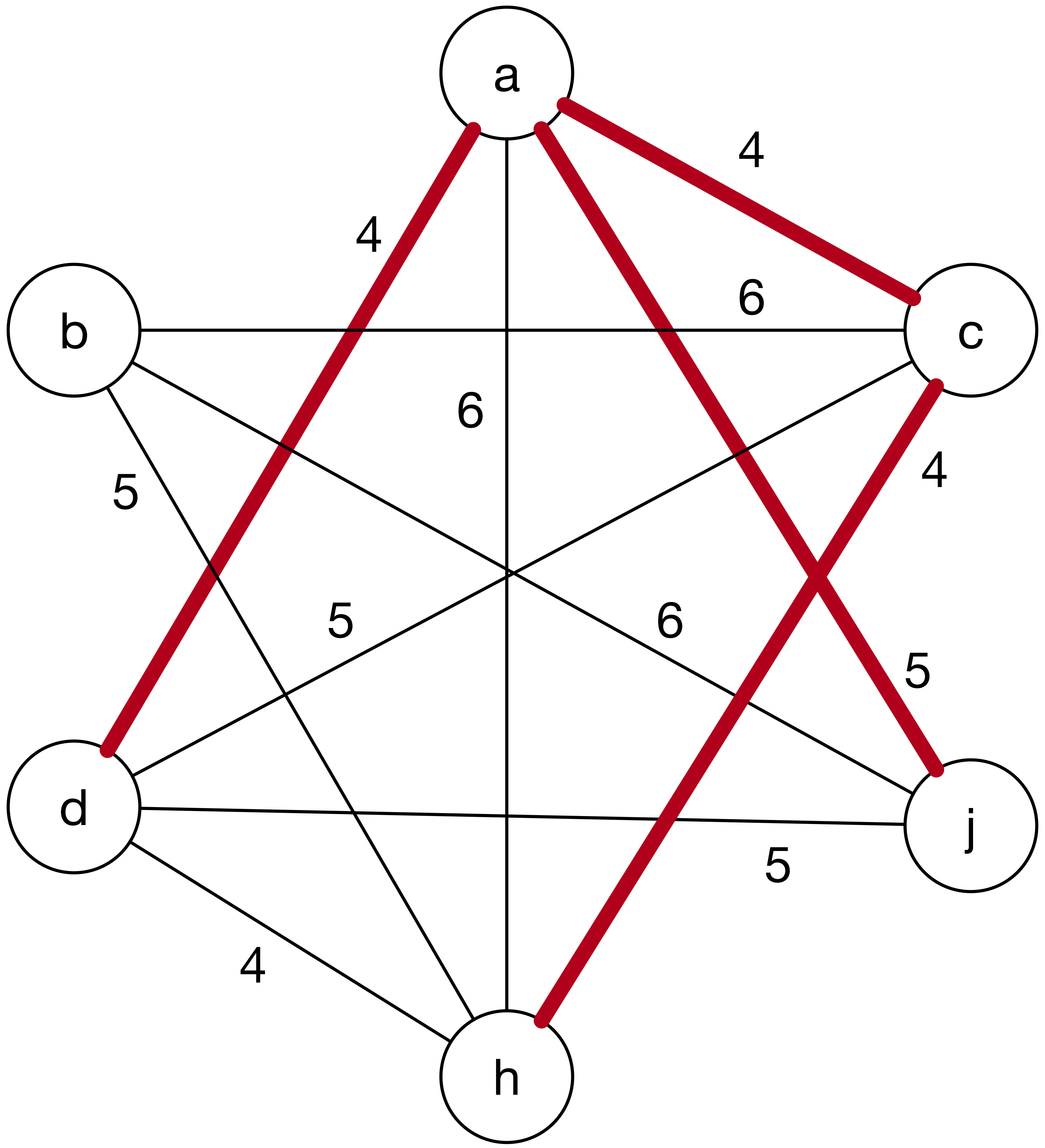
Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
~~dh~~	~~4~~
aj	5
dj	5
cd	5
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

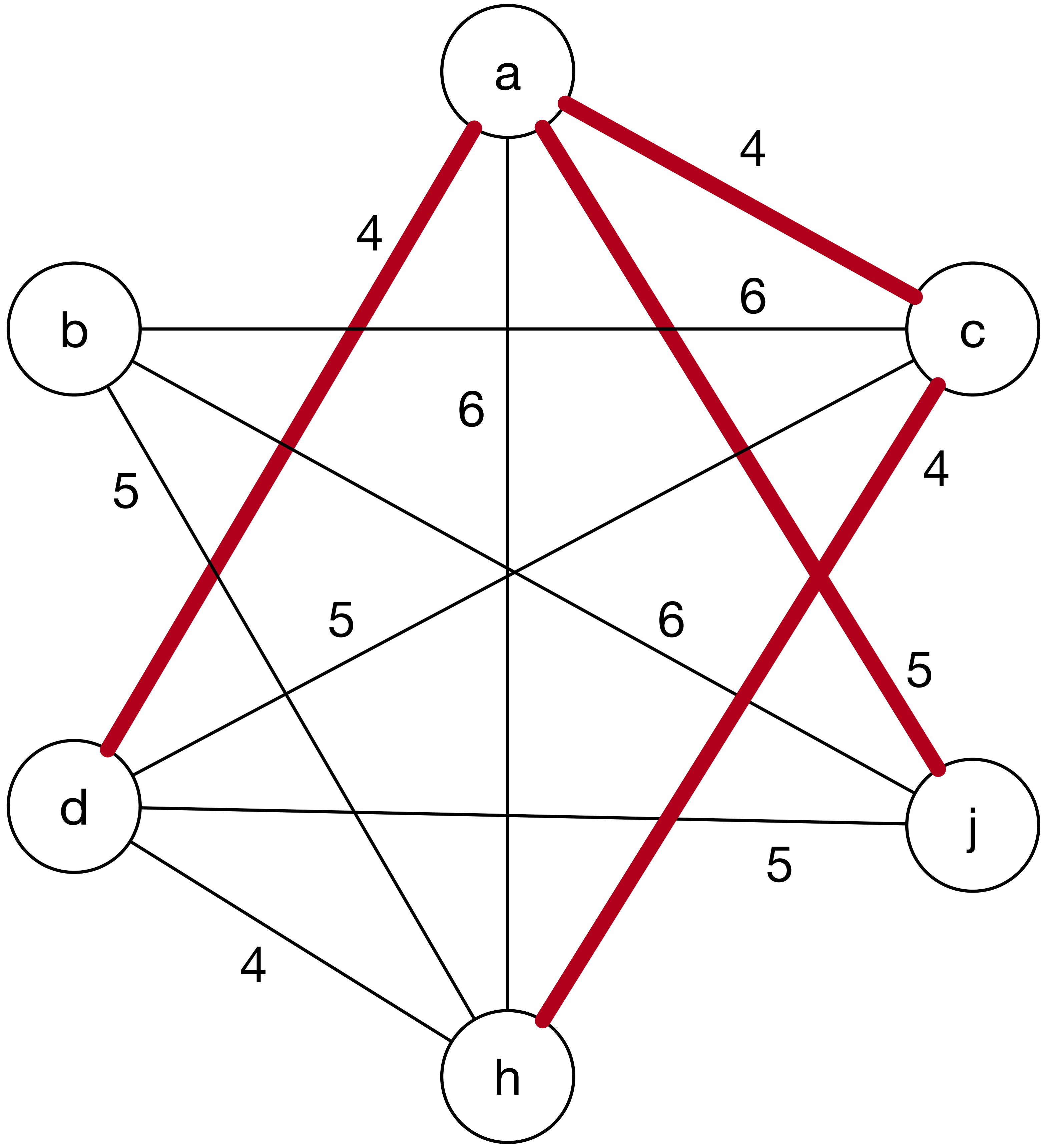
Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
~~dh~~	~~4~~
aj	5
~~dj~~	~~5~~
cd	5
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

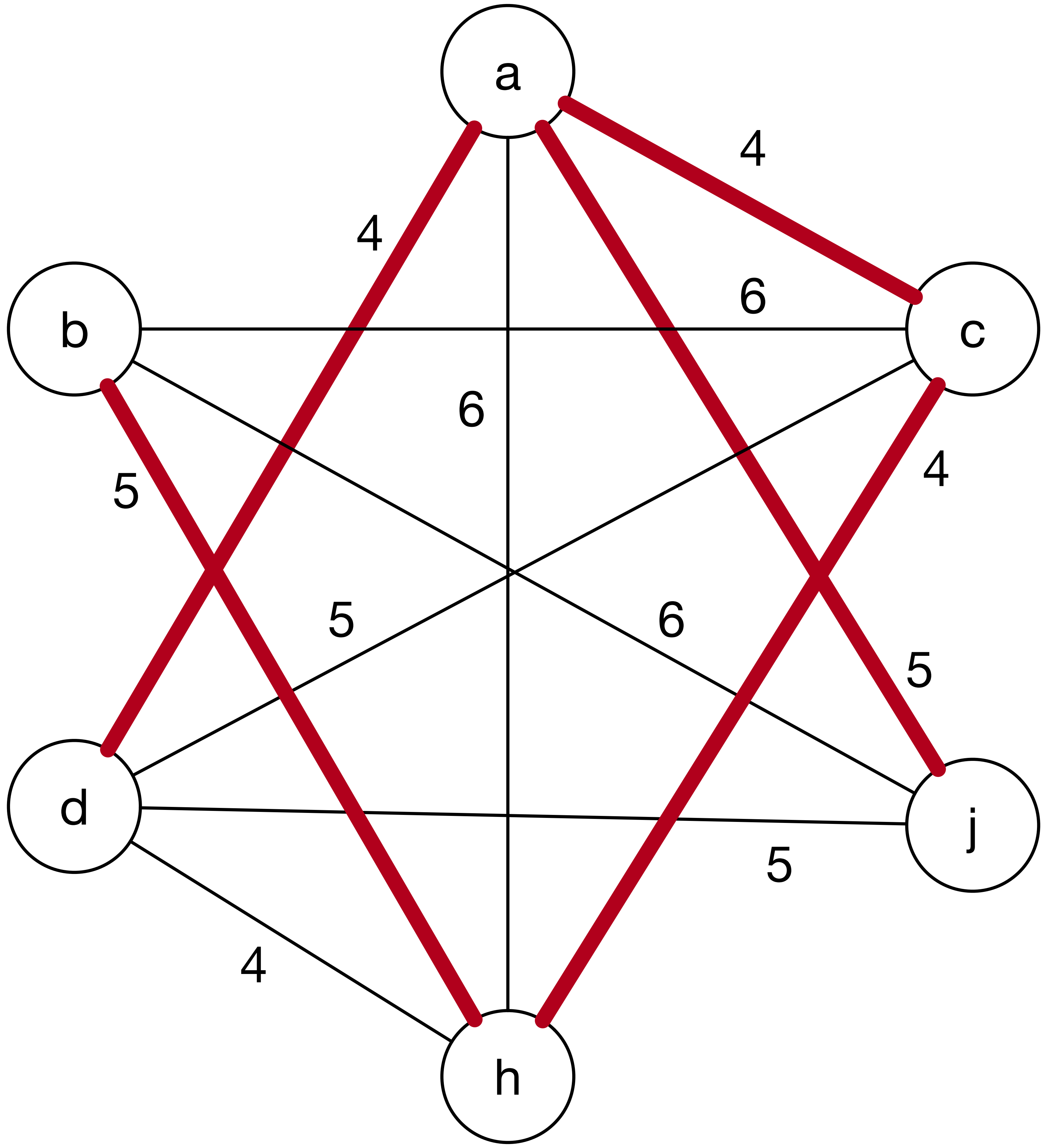
Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
~~dh~~	~~4~~
aj	5
~~dj~~	~~5~~
~~cd~~	~~5~~
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

Kruskal's algorithm



edge	weight
ac	4
ad	4
ch	4
~~dh~~	~~4~~
aj	5
~~dj~~	~~5~~
~~cd~~	~~5~~
bh	5
bc	6
bj	6
ah	6

Minimum spanning tree

Kruskal's algorithm, given some connected, weighted graph G=(V, E)

```
function kruskals(G)
    let E’ = {}
    sort edges in E   // can sort a vector or use min heap
    while |E’| < |V| - 1
        if next edge in sort (or heap) does not create cycle
            add this edge to E’
    return (V, E’)
```

Minimum spanning tree

Kruskal's algorithm, given some connected, weighted graph G=(V, E)

```
function kruskals(G)
    // using disjoint sets, find and union
    let E' = {}
    sort edges in E   // can sort a vector or use min heap
    make singleton sets of all nodes in V
    for each edge, (u,v) in E
        if find(u) != find(v)
            add edge to E'
            union(u, v)
    return (V, E')
```

Minimum spanning tree

Kruskal's algorithm

Is it greedy?

Minimum spanning tree

Kruskal's algorithm

Is it greedy? Yes!

Minimum spanning tree

Kruskal's algorithm

Is it greedy? Yes!

Worst-case complexity?

Minimum spanning tree

Kruskal's algorithm

Is it greedy? Yes!

Worst-case complexity? $\mathcal{O}(|E| \log |E|)$

The biggest cost is in sorting the edges by weights