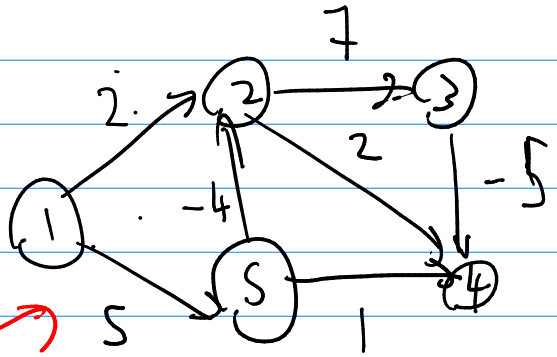
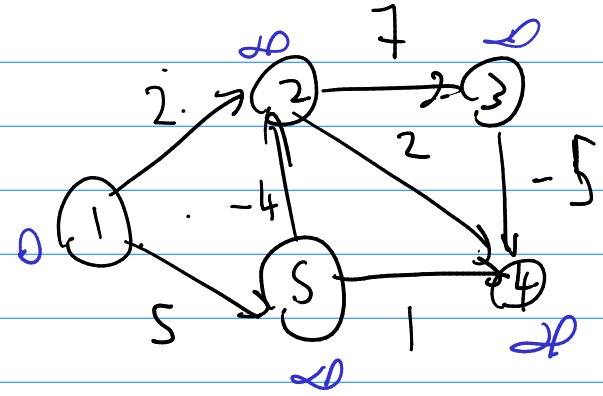
```
algorithm Dijkstra;
begin
    S := ∅; S̄ := N;
    d(i) := ∞ for each node i ∈ N;
    d(s) := 0 and pred(s) := 0;
    while |S| < n do
    begin
        let i ∈ S̄ be a node for which d(i) = min{d(j) : j ∈ S̄};
        S := S ∪ {i};
        S̄ := S̄ − {i};
        for each (i, j) ∈ A(i) do
            if d(j) > d(i) + c_ij then d(j) := d(i) + c_ij and pred(j) := i;
    end;
end;
```

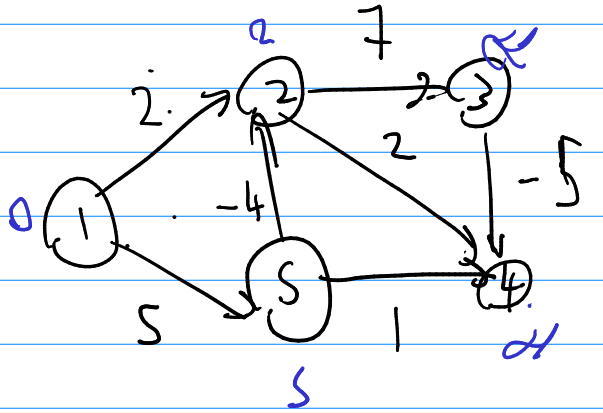
Figure 4.6 Dijkstra's algorithm.

Dijkstra can not solve this →

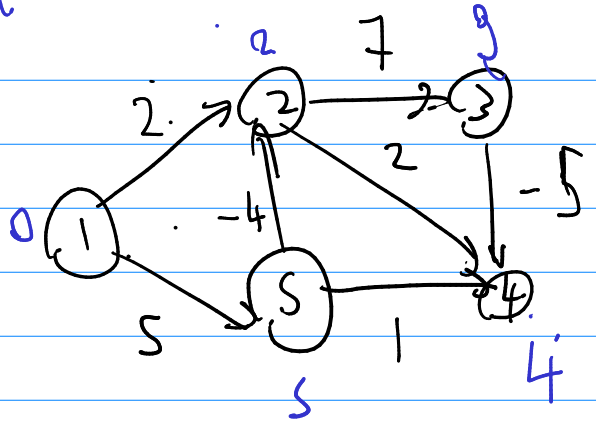



```
algorithm modified label-correcting;
begin
    d(s) := 0 and pred(s) := 0;
    d(j) := ∞ for each node j ∈ N − {s};
    LIST := {s};
    while LIST ≠ ∅ do
    begin
        remove an element i from LIST;
        for each arc (i, j) ∈ A(i) do
        if d(j) > d(i) + c_ij then
        begin
            d(j) := d(i) + c_ij;
            pred(j) := i;
            if j ∉ LIST then add node j to LIST;
        end;
    end;
end;
```



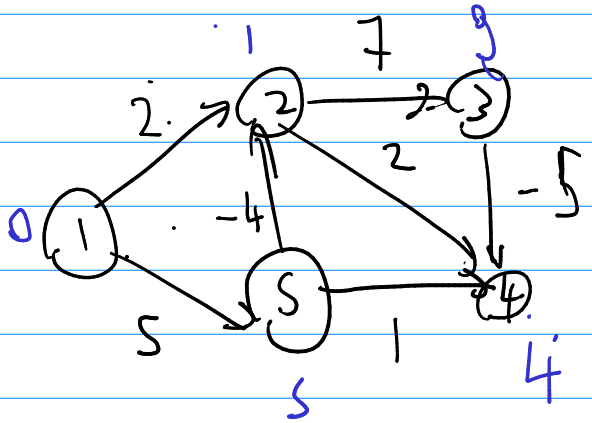


i = 2



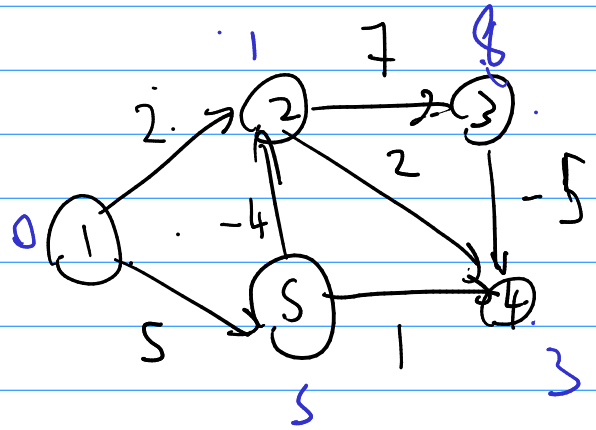
LIST = {5, 3, 4}

i = 5



LIST = {3, 4, 2}

i = 2



LIST = {3, 4}

LIST will be exhausted. We are done here.

Negative Cycles