2.20 Disenya un algorisme de cost raonable per trobar els estats no accesibles d'un DFA d'entrada.

Sigui $D = (Q, \Sigma, \delta, q_0, F)$ el DFA d'entrada, on:

- Q, conjunt d'estats
- Σ, alfabet
- $\delta : Q \times \Sigma \to Q$, funció de transició
- q_0, estat inicial
- F, conjunt d'estats finals (o acceptadors)

Algorisme

```
c = new Queue<>();
c.push(q_0);
while (not c.empty()) {
    q = c.pop();
    for (s in Σ) {
        q' = δ(q, s);
        if (q' ∉ A) {
            A = A ∪ {q'};
            c.push(q');
        }
    }
}
return Q \ A
```

L'algorisme utilitza en un BFS per trobar els estats accessibles i retornar els no accessibles.

Cost

El cost serà el d'executar un BFS al graf de l'autòmat, que té $|Q|$ nodes (o estats) i $|Q| \cdot |\Sigma|$ arestes (o transicions):

$$O(D) = O(|Q| + |Q| \cdot |\Sigma|) \in O(|Q| \cdot |\Sigma|)$$

> L'algorisme es podria millorar usant heurístiques o poda.