Bucket queue 10 poeng

En *bucket queue* (eller en bøttekø) er en datastruktur som er inspirert av *bucket sort* og kan brukes som en *prioritetskø*.

I en bucket queue kan elementer bli lagt til med en prioritet mellom 0 og $N-1$, der N er et positivt heltall som fastsettes når køen opprettes. Ved et kall på `RemoveMin`, så vil et element med lavest mulig prioritet fjernes og returneres; dersom det er flere elementer med samme prioritet spiller det ikke noen rolle hvilket element som fjernes og returneres.

(a) Forklar *kort* hvilken datastruktur som passer godt for en bucket queue.

(b) Gi en algoritme for `Insert` for en bucket queue.

```
Input: En bucket queue Q med prioriteter fra 0 til N − 1, og et element x med prioritet p
Output: Q med x satt inn med prioritet p
1 Procedure Insert(Q, x, p)
  |  // ···
```

(c) Gi en algoritme for `RemoveMin` for en bucket queue. Du kan anta at køen ikke er tom.

```
Input: En bucket queue Q med prioriteter fra 0 til N − 1
Output: En x med lavest mulig prioritet, som er fjernet fra Q
1 Procedure RemoveMin(Q)
  |  // ···
```

(d) Anta at $N = 100$. Oppgi kjøretidskompleksiteten på `Insert` og `RemoveMin` for en kø med n elementer.

(a) 1 poeng. Siden vi har prioriteter fra 0 til et kjent maksimum N, passer et enkelt array av bøtter perfekt. Bøttene kan for eksempel være en enkeltlenket liste. Dette er helt tilsvarende bucket sort.

(b) Opp til 3 poeng. Legg til x i bøtta på plass p i Q.

```
Input: En bucket queue Q med prioriteter fra 0 til N − 1, og et element x med prioritet
       p
Output: Q med x satt inn med prioritet p
1 Procedure Insert(Q, x, p)
2  |  B ← Q[p]
3  |  add x to B
```

(c) Opp til 4 poeng. Finn det den første ikke-tomme bøtta, og fjern og returner det første i den bøtta.

```
Input: En bucket queue Q med prioriteter fra 0 til N − 1
Output: En x med lavest mulig prioritet, som er fjernet fra Q
1 Procedure RemoveMin(Q)
2  |  for i ← 0 to N − 1 do
3  |  |  B ← Q[i]
4  |  |  if B is not empty then
5  |  |  |  return B.pop()
```

(d) 1 poeng for hver av algoritmene. Siden N er konstant så er begge algoritmene $\mathcal{O}(1)$.

En besvarelse som tar utgangspunkt i en binær heap (eller andre mindre egnede strukturer) kan få maksimalt 4 poeng.