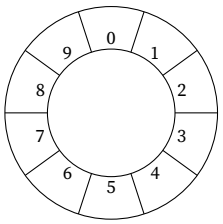
Separate chaining 10 poeng

(a) Vi starter med et tomt array på størrelse $N = 10$.



Hashfunksjonen du skal bruke er $h(k, N) = k \mod N$, som for dette eksempelet blir $h(k, 10) = k \mod 10$. Altså hasher et tall til sitt siste siffer.

Bruk *separate chaining* til å sette inn disse nøkkel- og verdiparene i den gitte rekkefølgen:

$$(21, a), (54, b), (82, c), (10, d), (20, e), (44, f), (10, g)$$

Skriv ned resultatet etter siste innsetting, med hver linje på formen:

$$i\colon (k_1, v_1), (k_2, v_2), \ldots$$

der i er en indeks i arrayet, og $(k_1, v_1), (k_2, v_2), \ldots$ er nøkkel- og verdipar.

(b) Forklar kort hvordan algoritmen for innsetting ved separate chaining fungerer, og skisser algoritmen med pseudokode. Du kan anta at du har en hashfunksjon h slik at $h(k, N)$ gir et tall mellom 0 og $N - 1$ for en vilkårlig nøkkel k. Du trenger ikke ta høyde for rehashing.

```
Input: Et array A av størrelse N, en nøkkel k og verdi v
Output: Et array der (k, v) er satt inn med separate chaining
1 Procedure SeparateChainingInsert(A, k, v)
  | // ...
```

(c) Redegjør kort for kjøretidskompleksiteten til innsetting ved separate chaining.

(a) Resultatet av innsettingen ser slik ut:

$$
\begin{aligned}
&0\colon (10, g), (20, e)\\
&1\colon (21, a)\\
&2\colon (82, c)\\
&3\colon\\
&4\colon (54, b), (44, f)\\
&5\colon\\
&6\colon\\
&7\colon\\
&8\colon\\
&9\colon
\end{aligned}
$$

- 3 poeng for riktig svar.
- Det trekkes 1.5 poeng for å ikke overskrive en lik nøkkel med ny verdi.

(b) Det viktigste er at studenten formidler en god forståelse av hvordan separate chaining fungerer.

Innsetting med separate chaining tar utgangspunkt i å bruke hashen av nøkkelen som indeks, og plassere nøkkel-/verdi-paret i en bøtte (som regel implementert som en lenket liste). Dersom nøkkel-/verdi-paret allerede finnes i listen, skal verdien erstattes.

I implementasjonen under antar vi lister som er implementert som dynamiske arrayer.

```
Input: Et array A av størrelse N, en nøkkel k og verdi v
Output: Et array der (k, v) er satt inn med separate chaining
1  Procedure SeparateChainingInsert(A, k, v)
2      i ← h(k, N)
3      B ← A[i]
4      if B = null then
5          A[i] ← a list containing the element (k, v)
6          return A
7      for j ← 0 to |B| − 1 do
8          if key of B[j] is k then
9              B[j] ← (k, v)
10             return A
11     append (k, v) to B
12     return A
```

- 5 poeng for en god forklaring med pseudokode.
- Det gis full uttelling for selvforklarende eller godt kommentert pseudokode.
- Det gis inntil 3 poeng dersom pseudokoden mangler.
- Det trekkes 1 poeng for å ikke overskrive en lik nøkkel med ny verdi.
- Det trekkes 1 poeng dersom det ikke kommer klart frem hvordan bøttene initialiseres.

(c) Kjøretiden til separate chaining er $O(n)$ i verste tilfelle, men under antagelsen av en god hashfunksjon og en tilstrekkelig lav load factor vil lengden på hver liste være begrenset av en konstant, som gir forventet $O(1)$.

- 2 poeng.
- 1 poeng for kun verste tilfelle.
- 1 poeng for kun forventet kjøretidskompleksitet.