

29. Tenim un flux de dades que conté tots els nombres al conjunt $\{1, \dots, n\}$ menys un. Dissenyeu un algorisme per trobar l'element que falta. El vostre algorisme pot llegir el flux només una vegada i fer servir $O(\log n)$ espai a memòria. L'algorisme ha de proporcionar la resposta correcta després de processar tota l'entrada.

Per resoldre aquest problema, podem utilitzar l'algorisme de cerca binària per trobar l'element que falta en el flux de dades. L'idea és utilitzar la propietat que els nombres estan en ordre consecutiu i només hi falta un element.

Aquí tens un exemple d'algorisme que compleix amb els requisits especificats:

Inicialitza les variables esquerra i dreta amb els valors 1 i n respectivament, on n és la màxima possible del conjunt de nombres.

Mentre esquerra sigui menor o igual que dreta, executa el següent pas:

- a. Calcula el punt mig de esquerra i dreta com $\text{mig} = (\text{esquerra} + \text{dreta}) / 2$.
- b. Comprova si l'element mig és igual al valor esperat. Per fer això, pots llegir el nombre actual del flux de dades. Si són iguals, l'element que falta es troba a la dreta de mig, per tant actualitza $\text{esquerra} = \text{mig} + 1$. Si no són iguals, l'element que falta es troba a l'esquerra de mig, per tant actualitza $\text{dreta} = \text{mig} - 1$.

Un cop s'hagi completat el pas anterior, l'element que falta serà esquerra, ja que dreta haurà quedat igual a esquerra - 1.

Aquest algorisme utilitza la cerca binària per reduir a la meitat el rang de possibles valors en cada iteració. Això implica que només es necessiten $O(\log n)$ iteracions per trobar l'element que falta. A més, només utilitza $O(\log n)$ espai a memòria per mantenir les variables esquerra, dreta i mig.

Cal tenir en compte que aquest algorisme assumeix que el flux de dades està ordenat i que només hi falta un element. Si aquestes suposicions no es compleixen, l'algorisme pot no donar la resposta correcta.