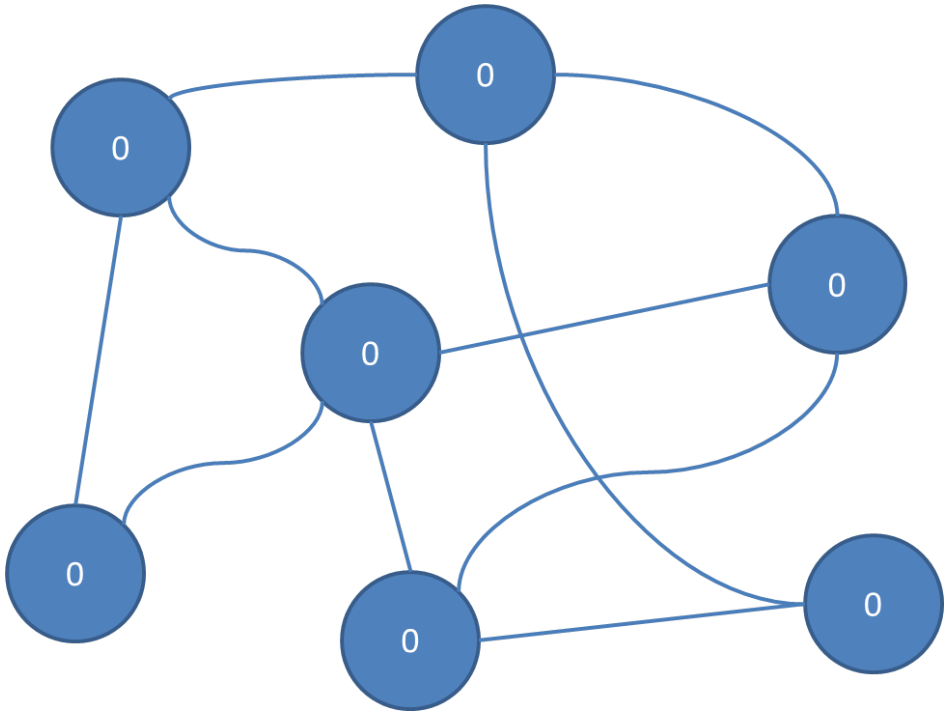
Another example

Wenda Qiu

2016 年 12 月 25 日

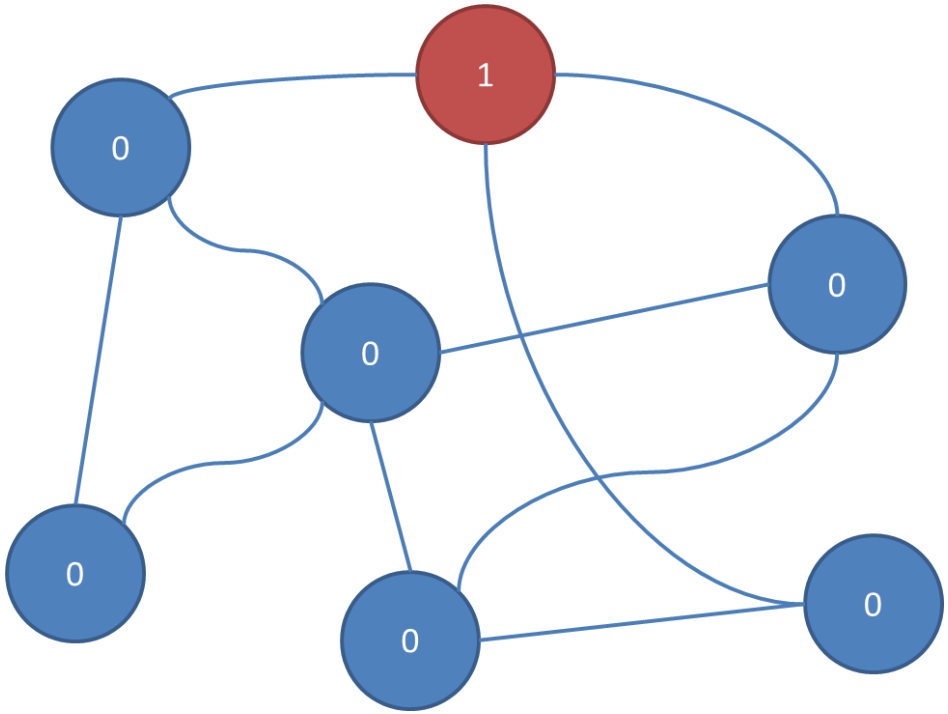
Another Problem

- an undirected graph



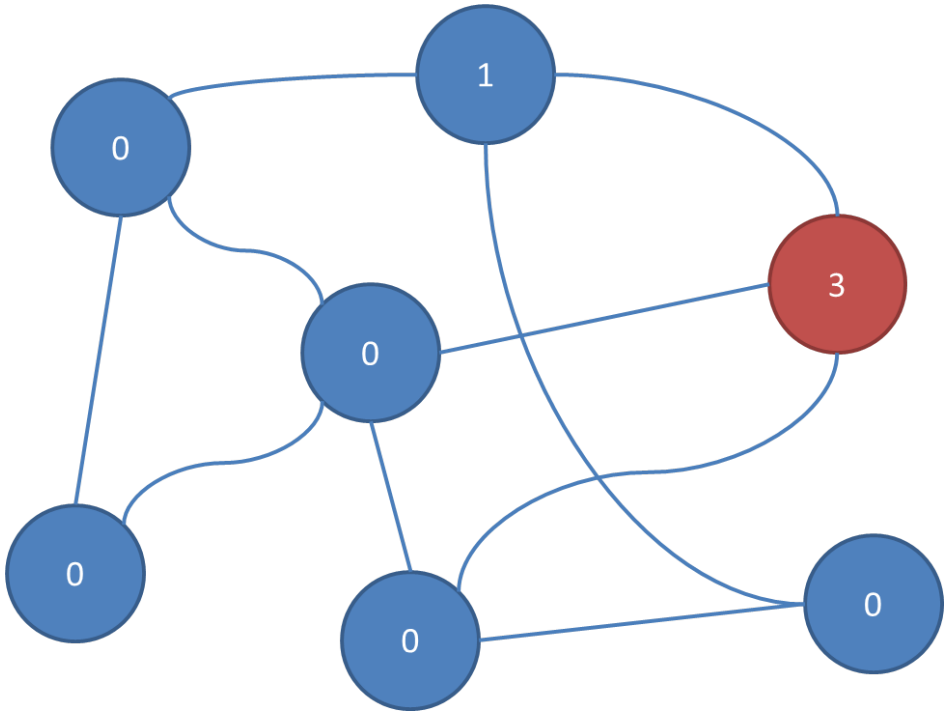
Another Problem

- an undirected graph
- add x to u



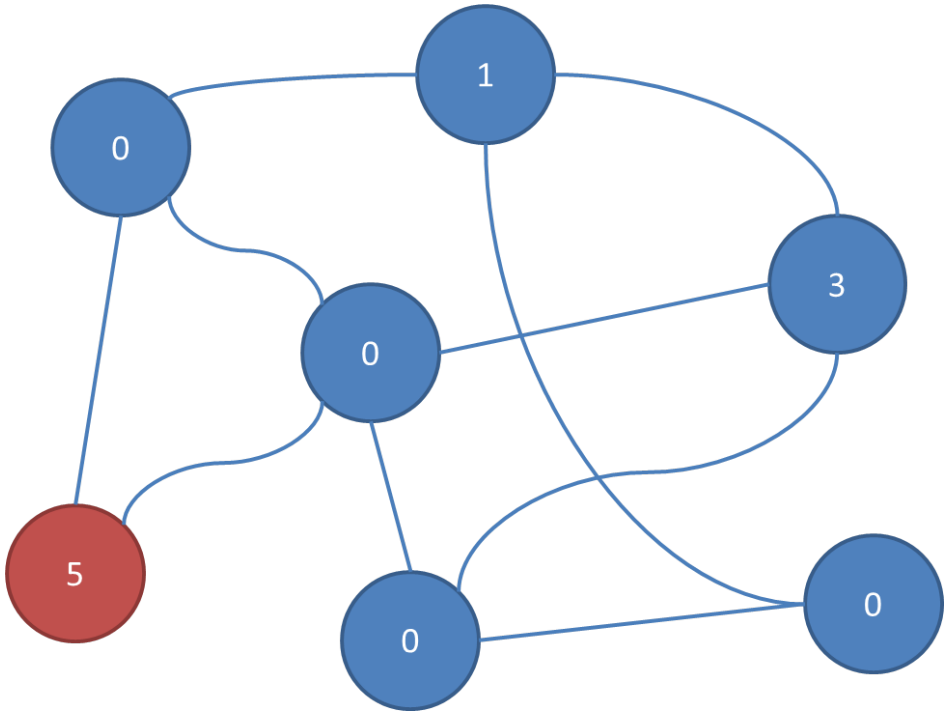
Another Problem

- an undirected graph
- add x to u



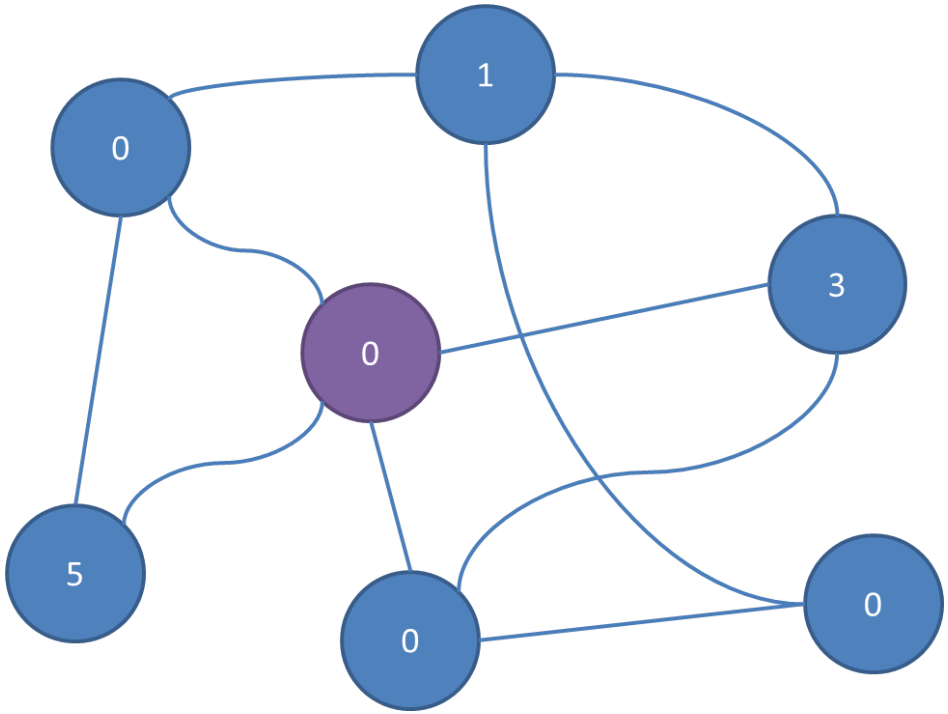
Another Problem

- an undirected graph
- add x to u



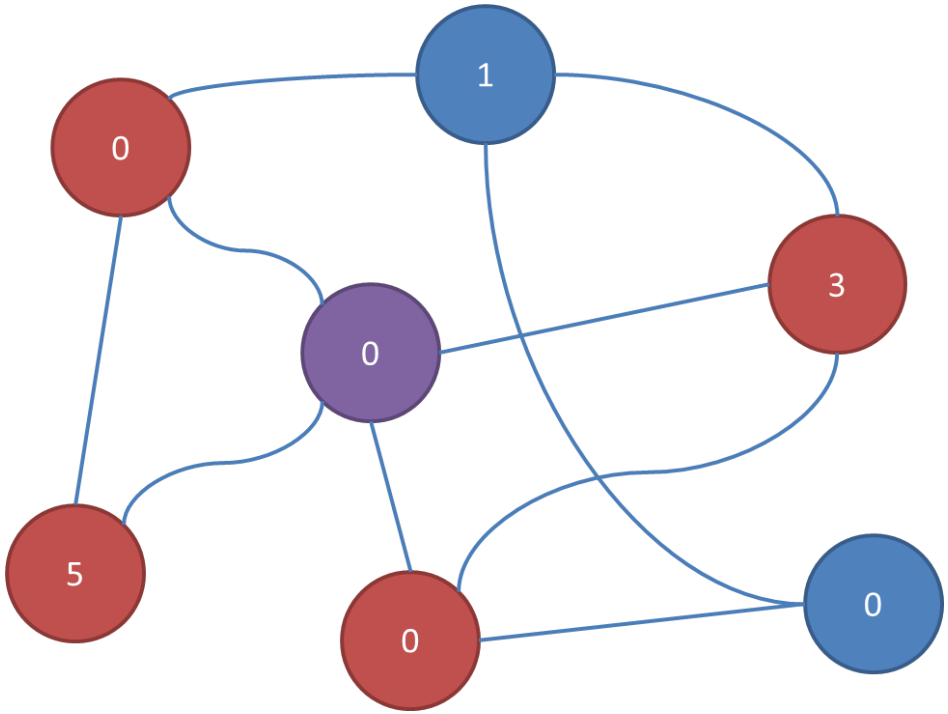
Another Problem

- an undirected graph
- add x to u
- query neighbors' sum



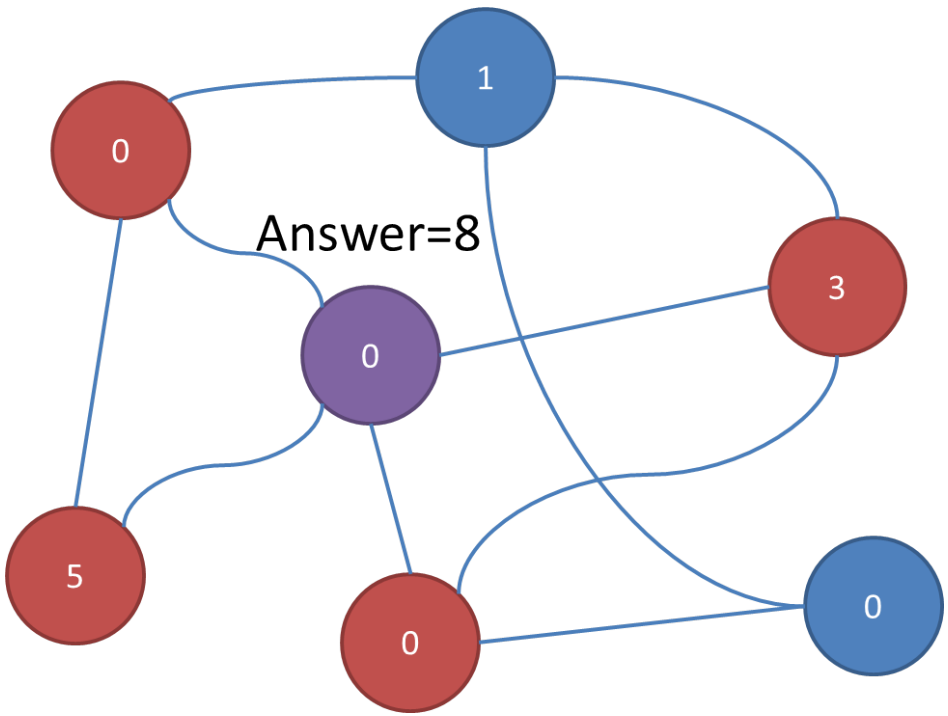
Another Problem

- an undirected graph
- add x to u
- query neighbors' sum



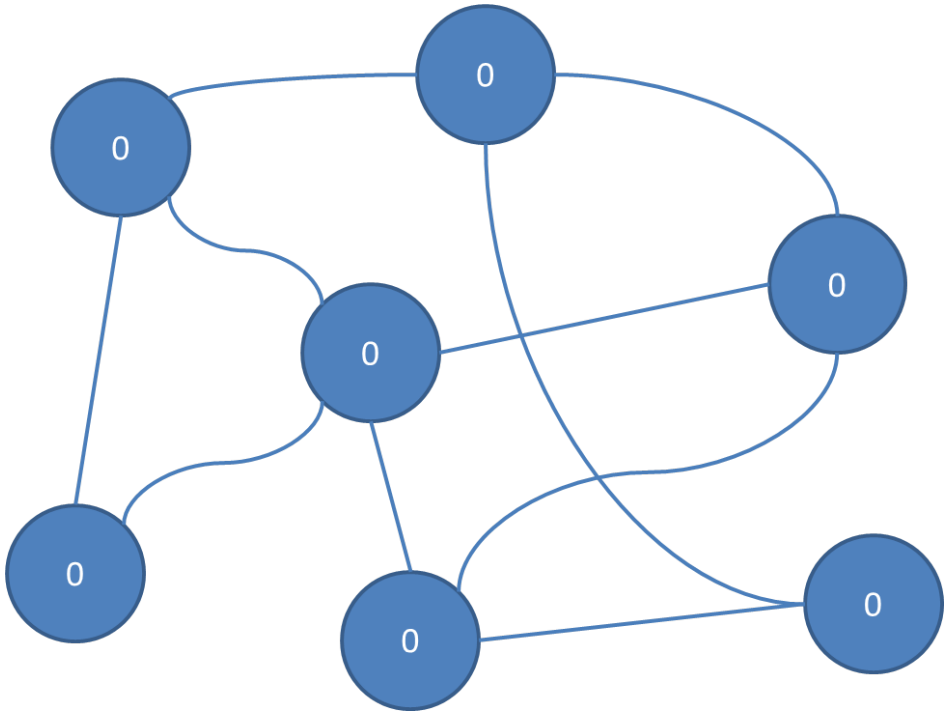
Another Problem

- an undirected graph
- add x to u
- query neighbors' sum
- $O(m) = n$



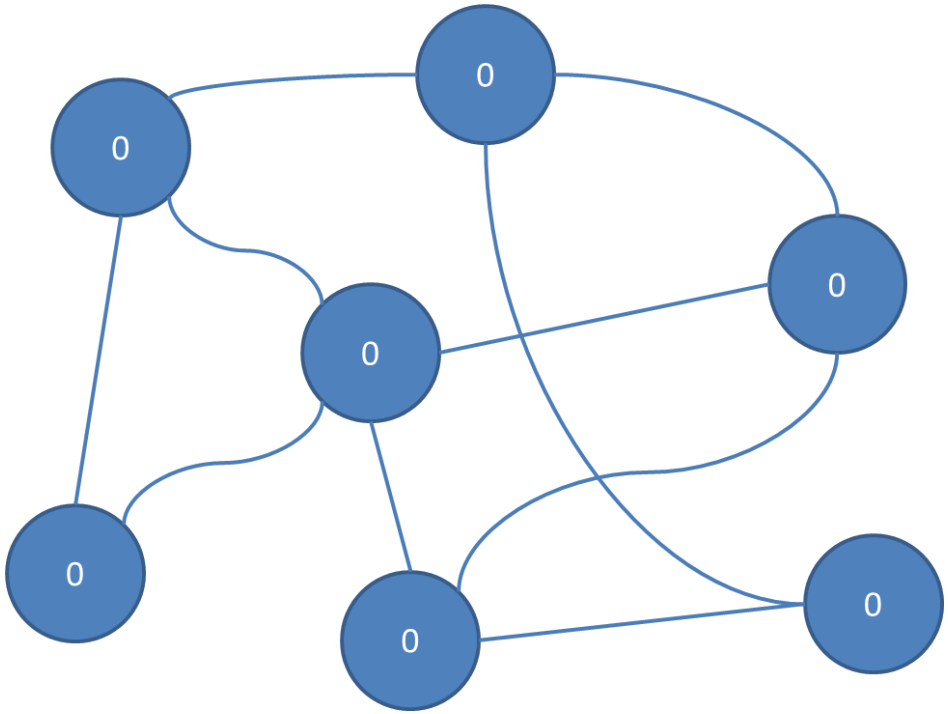
Naive Approach 1

- store the value $v[x]$



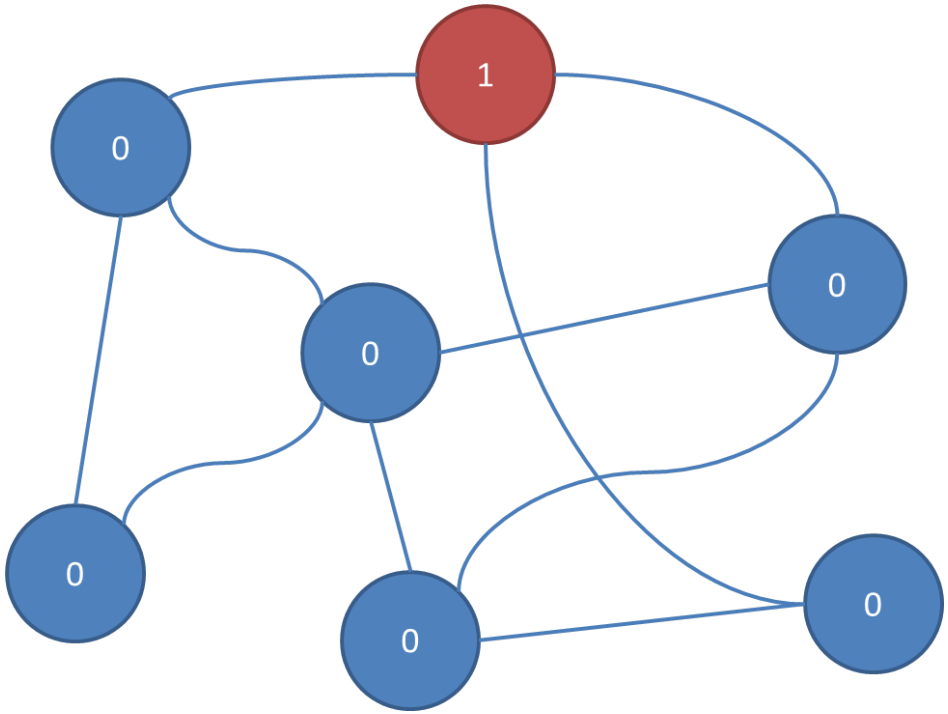
Naive Approach 1

- store the value $v[x]$
- $O(n)$ space



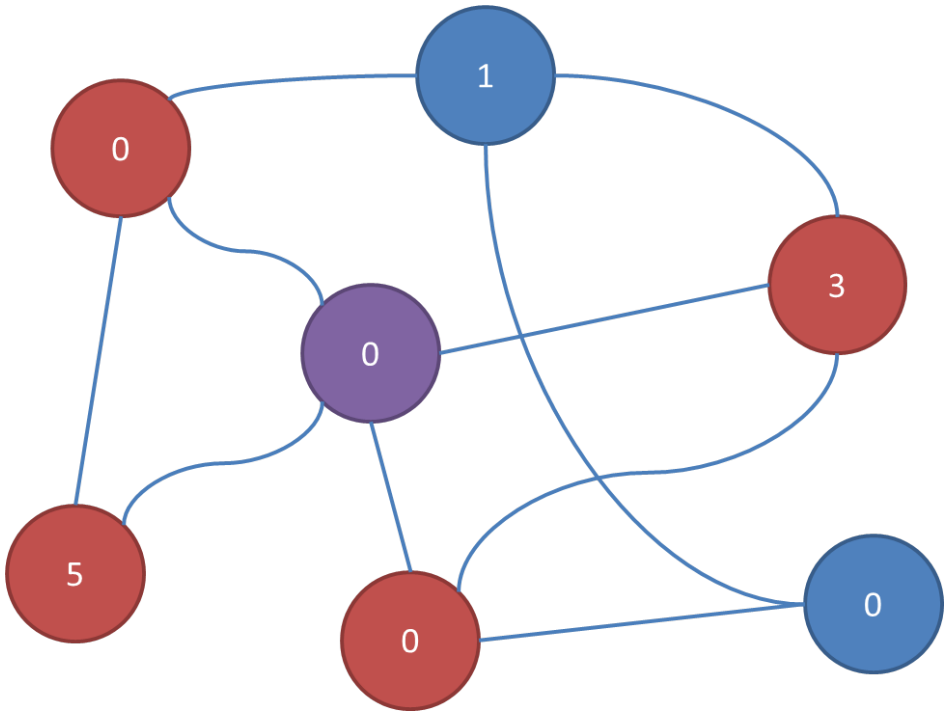
Naive Approach 1

- store the value $v[x]$
- $O(n)$ space
- $O(1)$ time for add



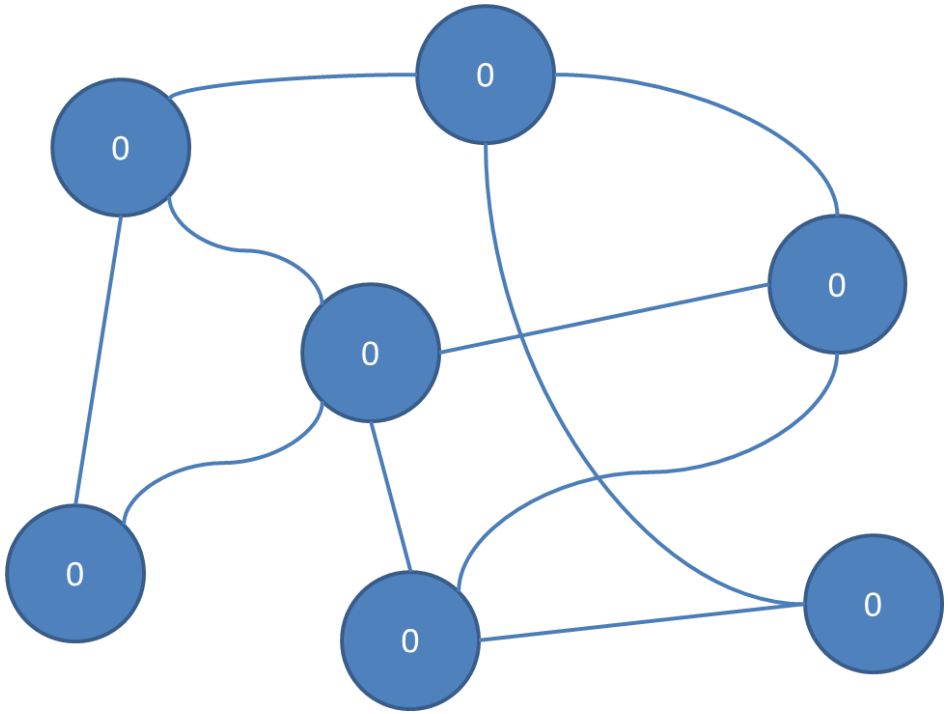
Naive Approach 1

- store the value $v[x]$
- $O(n)$ space
- $O(1)$ time for add
- $O(m)$ time for query



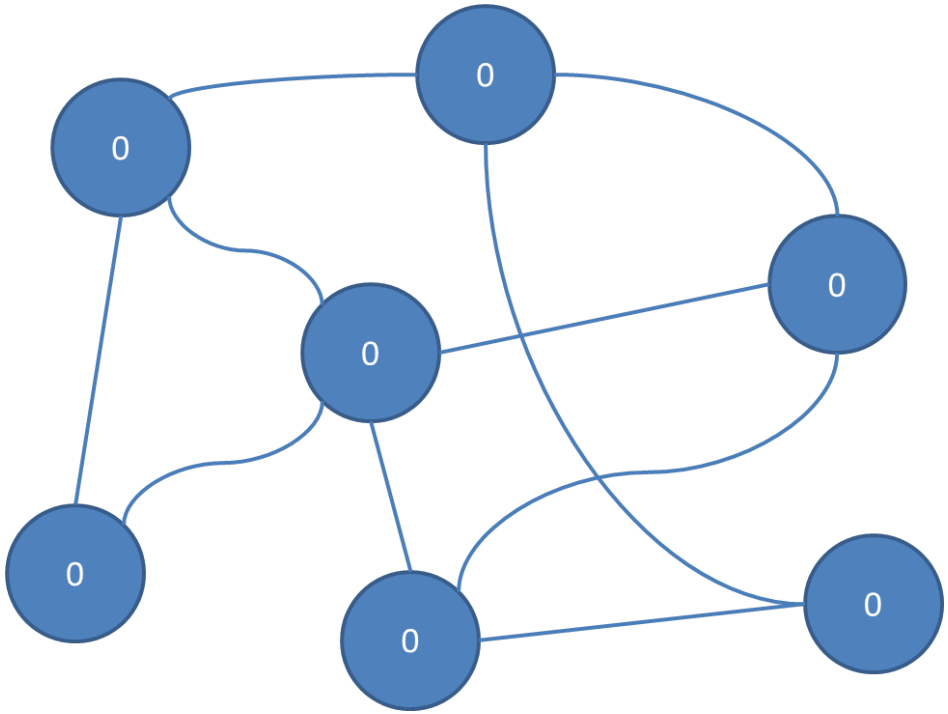
Naive Approach 2

- store the sum $s[x]$



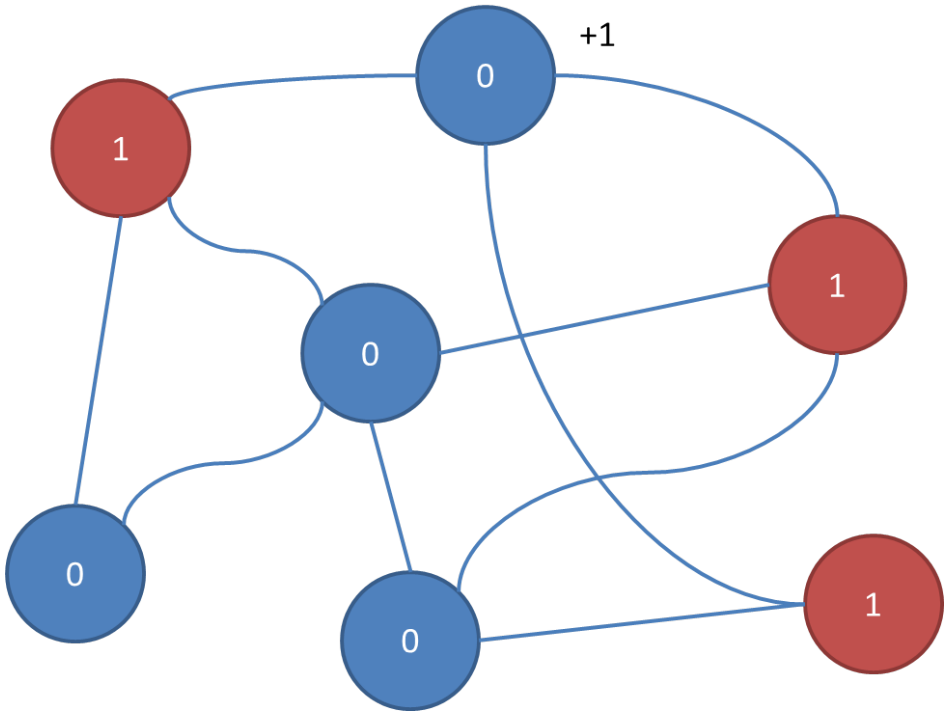
Naive Approach 2

- store the sum $s[x]$
- $O(n)$ space



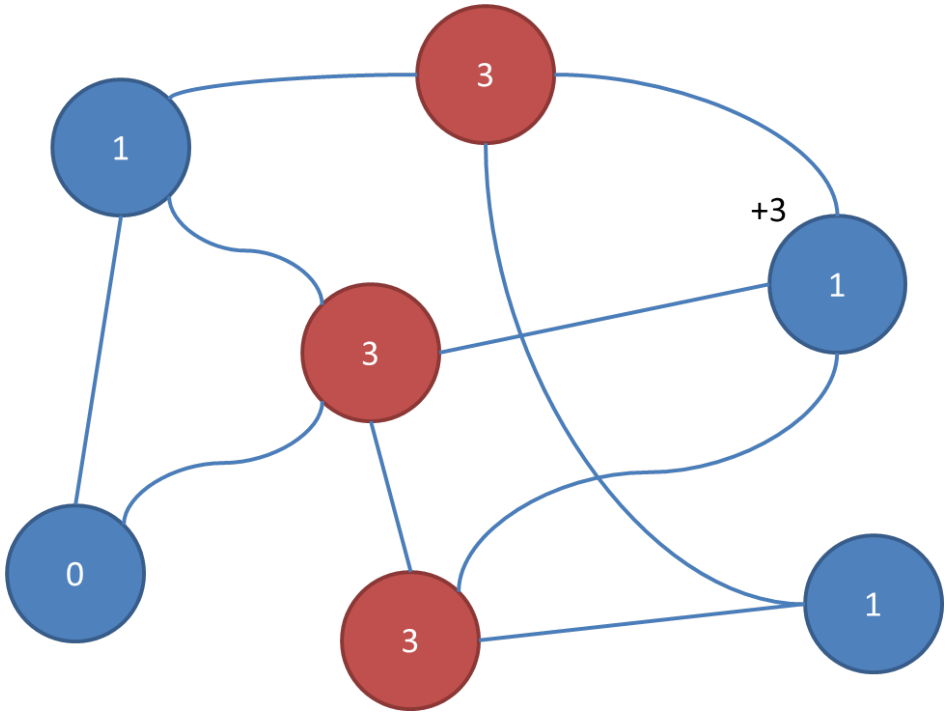
Naive Approach 2

- store the sum $s[x]$
- $O(n)$ space
- $O(n)$ time for add



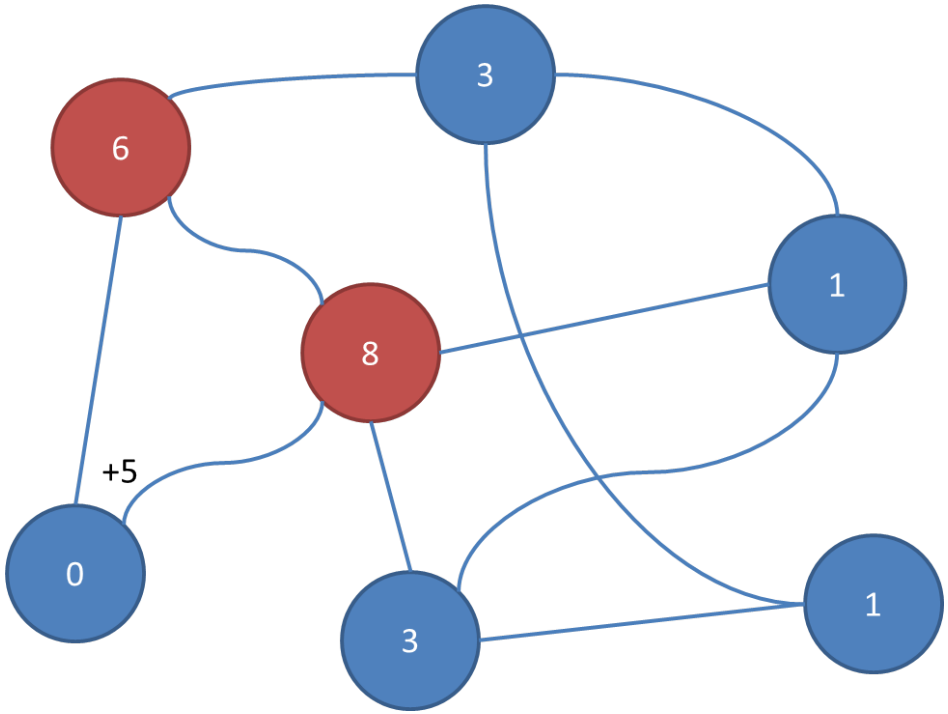
Naive Approach 2

- store the sum $s[x]$
- $O(n)$ space
- $O(n)$ time for add



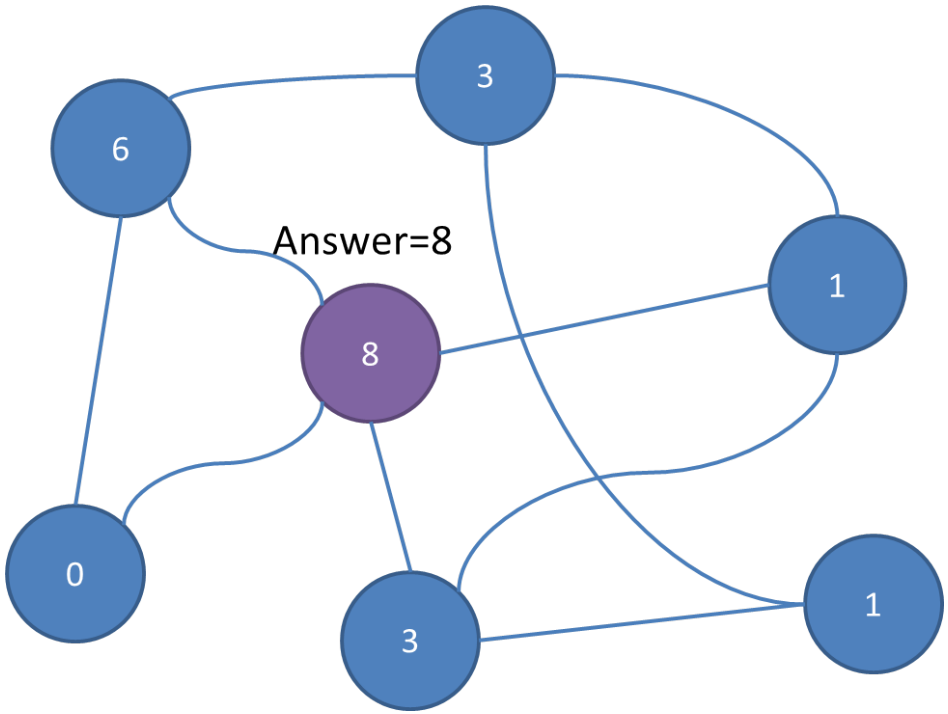
Naive Approach 2

- store the sum $s[x]$
- $O(n)$ space
- $O(n)$ time for add



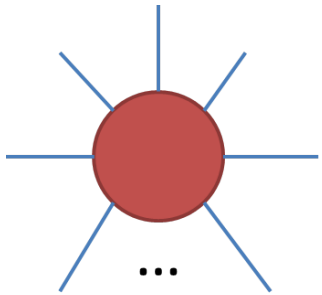
Naive Approach 2

- store the sum $s[x]$
- $O(n)$ space
- $O(n)$ time for add
- $O(1)$ time for query



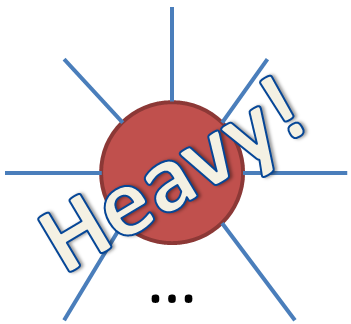
Observation

- bad when large $deg(x)$

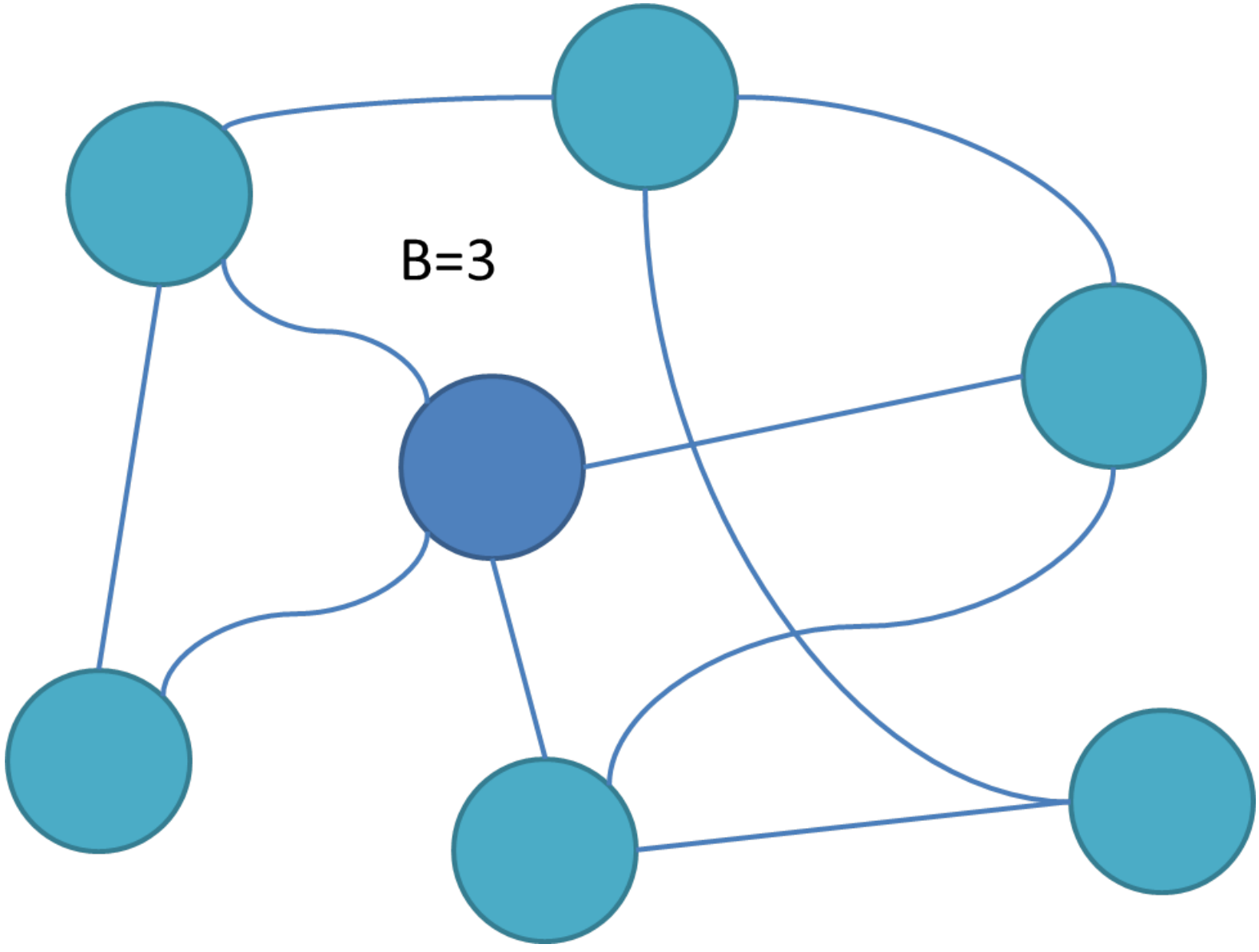


Observation

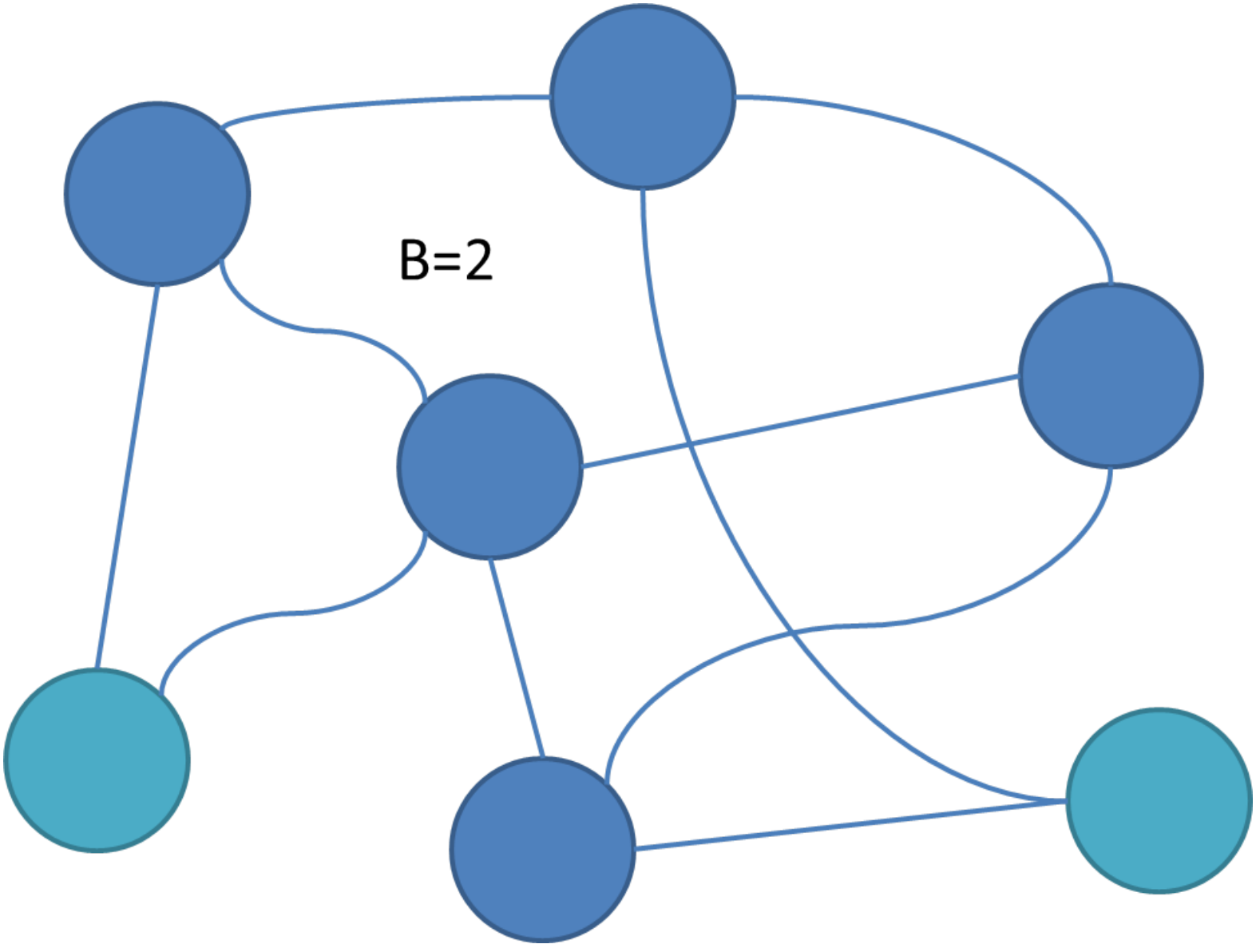
- bad when large $deg(x)$
- "heavy" when $deg(x) > B$



Heavy-Light Divide

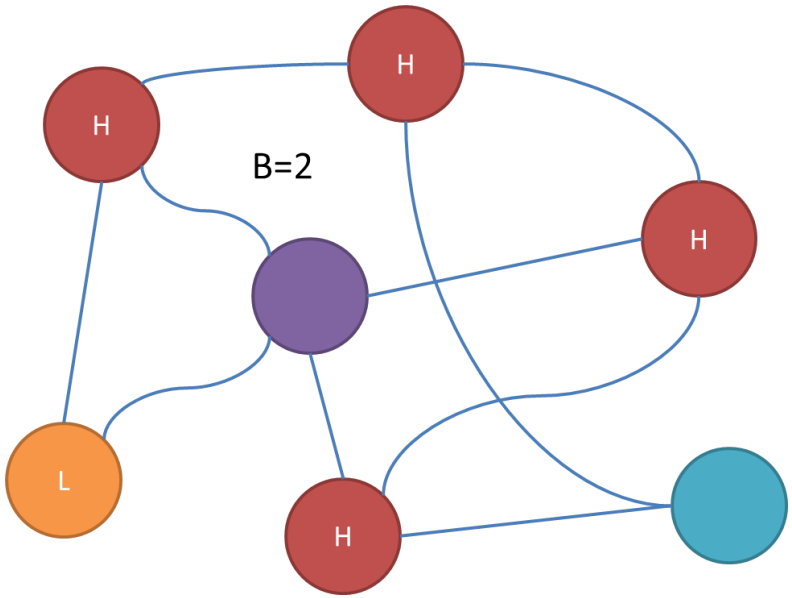


Heavy-Light Divide



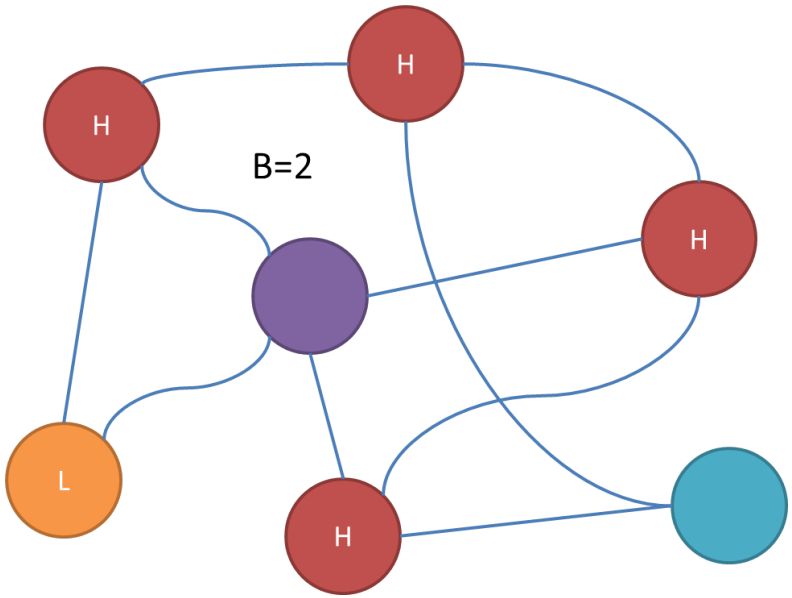
Combined Approach

- $s[x] = \sum v[\textit{heavy neighbors}] + \sum v[\textit{light neighbors}]$



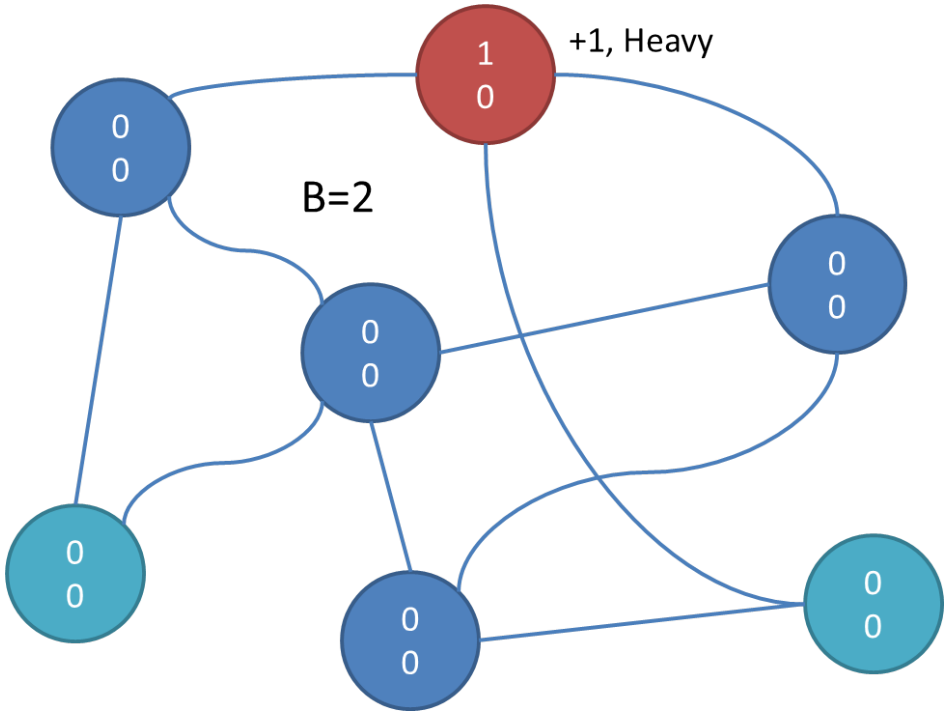
Combined Approach

- $s[x] = \sum v[\textit{heavy neighbors}] + \sum v[\textit{light neighbors}]$
- for $\sum v[\textit{heavy neighbors}]$ use approach 1
- for $\sum v[\textit{light neighbors}]$ use approach 2



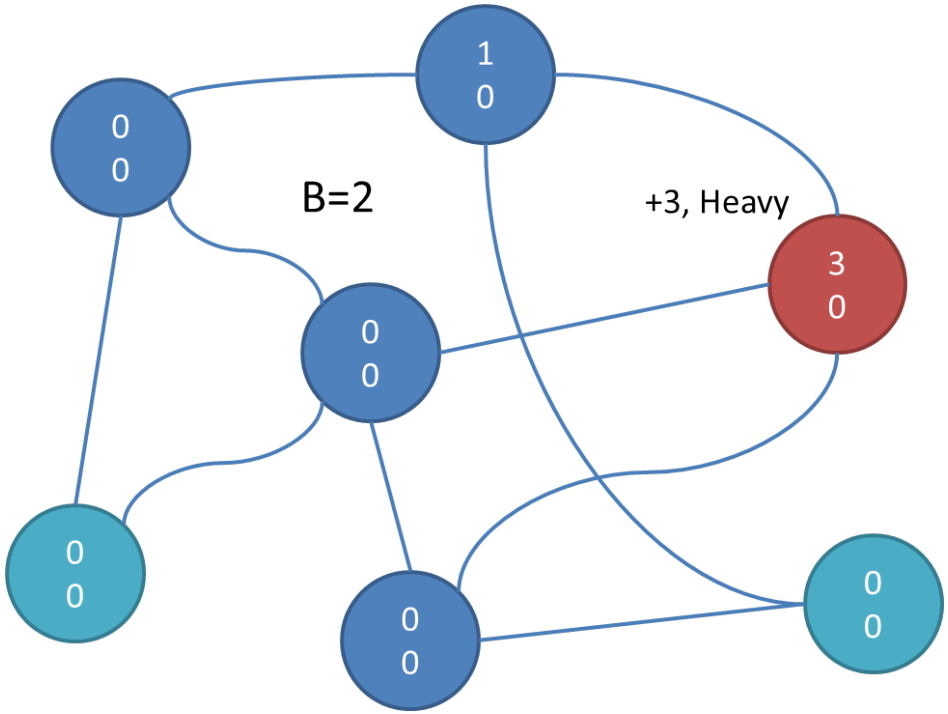
$Add(u, x)$ details

- if v is heavy, $vh[u] \leftarrow vh[u] + x$



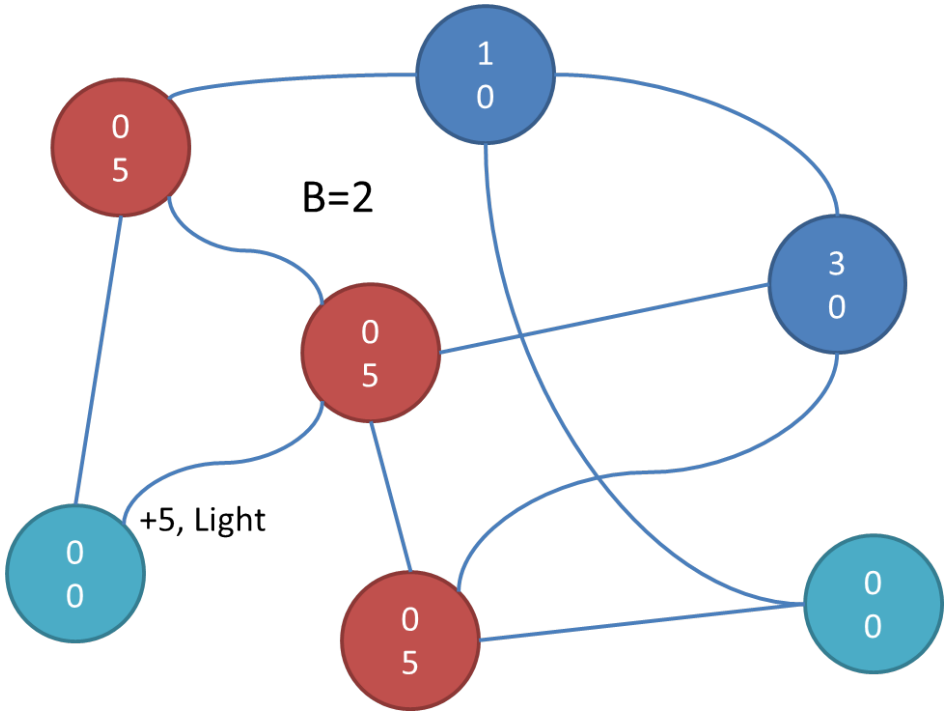
$Add(u, x)$ details

- if v is heavy, $vh[u] \leftarrow vh[u] + x$



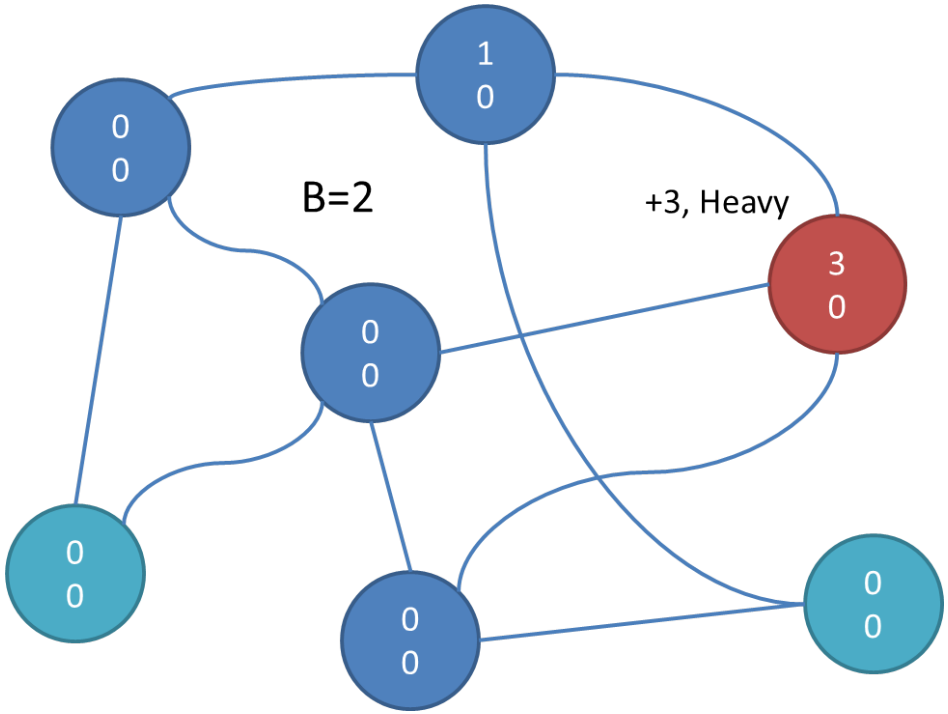
$Add(u, x)$ details

- if v is heavy, $vh[u] \leftarrow vh[u] + x$
- if u is light, $sl[v] \leftarrow sl[v] + x$, u, v are neighbors



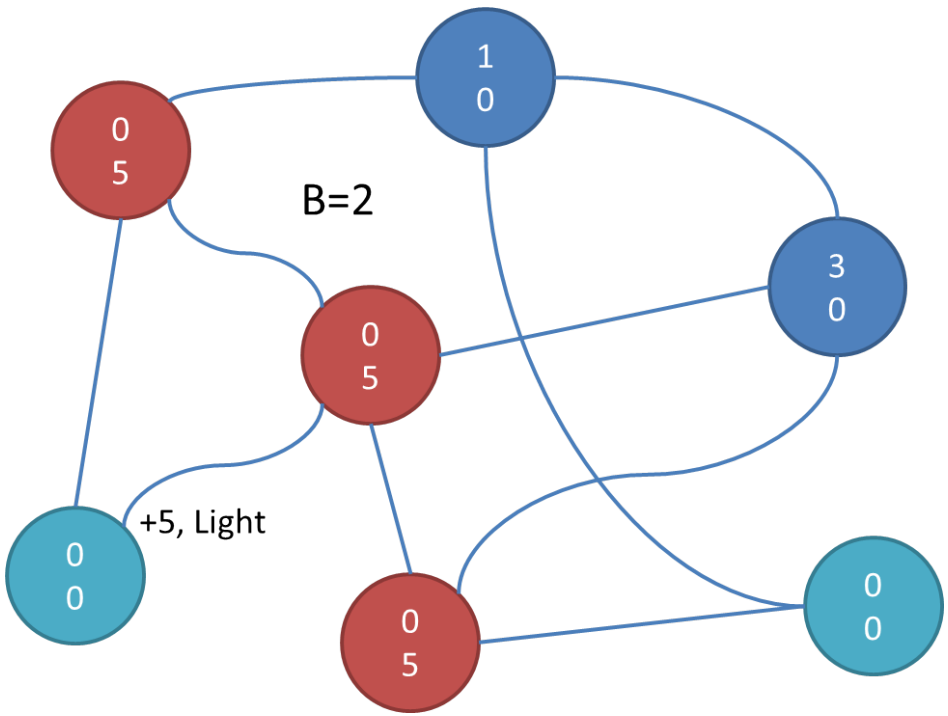
$Add(u, x)$ details

- if v is heavy, $vh[u] \leftarrow vh[u] + x$
- if u is light, $sl[v] \leftarrow sl[v] + x$, u, v are neighbors
- $O(1)$ time for heavy



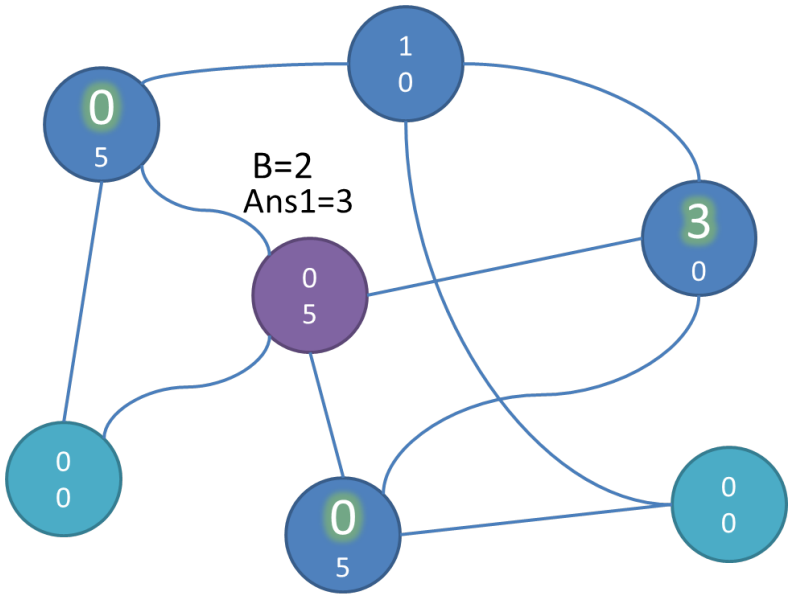
$Add(u, x)$ details

- if v is heavy, $vh[u] \leftarrow vh[u] + x$
- if u is light, $sl[v] \leftarrow sl[v] + x$, u, v are neighbors
- $O(1)$ time for heavy
- $O(B)$ time for light



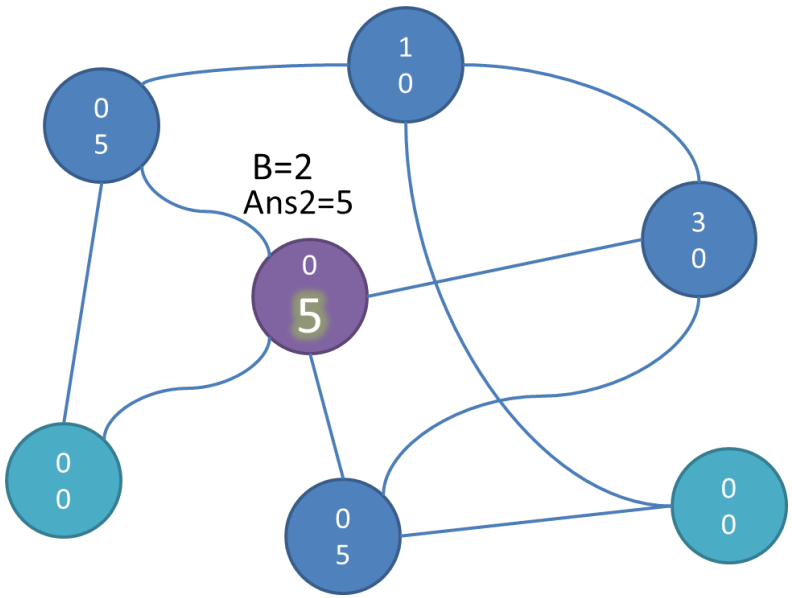
$Query(x)$ details

- $\sum v[heavy\ \ neighbors] = \sum_{y\ \ is\ \ heavy\ \ neighbor} vh[y]$



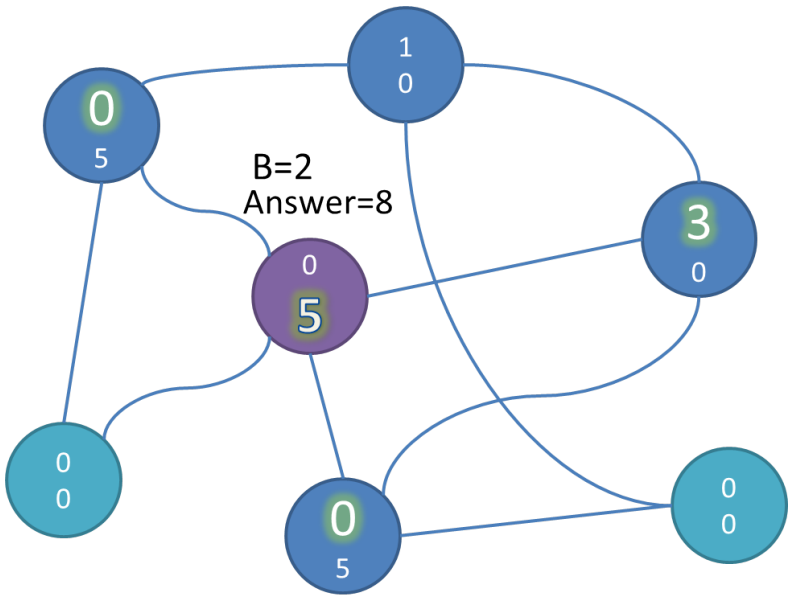
$Query(x)$ details

- $\sum v[heavy\ \ neighbors] = \sum_{y\ \ is\ \ heavy\ \ neighbor} vh[y]$
- $\sum v[light\ \ neighbors] = sl[x]$



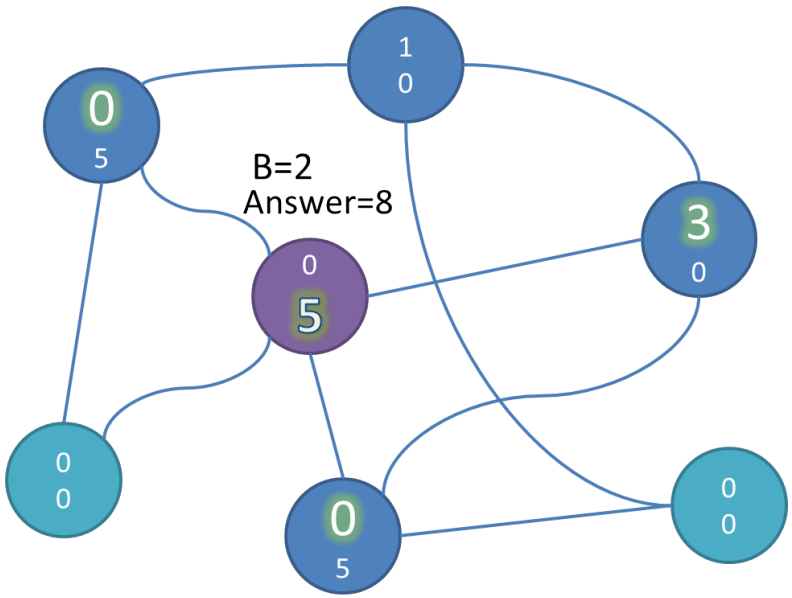
$Query(x)$ details

- $\sum v[heavy\ \ neighbors] = \sum_{y\ \ is\ \ heavy\ \ neighbor} vh[y]$
- $\sum v[light\ \ neighbors] = sl[x]$



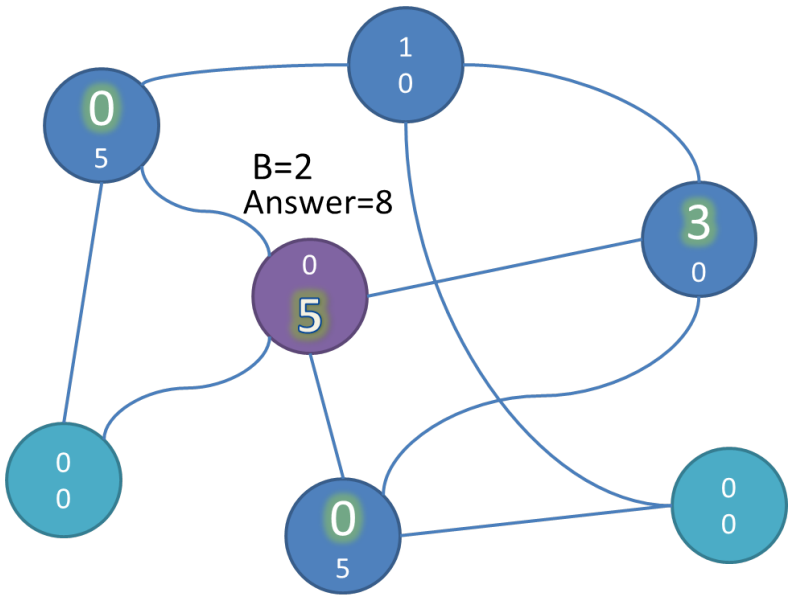
$Query(x)$ details

- $\sum v[heavy\ \ neighbors] = \sum_{y\ \ is\ \ heavy\ \ neighbor} vh[y]$
- $\sum v[light\ \ neighbors] = sl[x]$
- $O(1 + cnt\_heavy)$ time



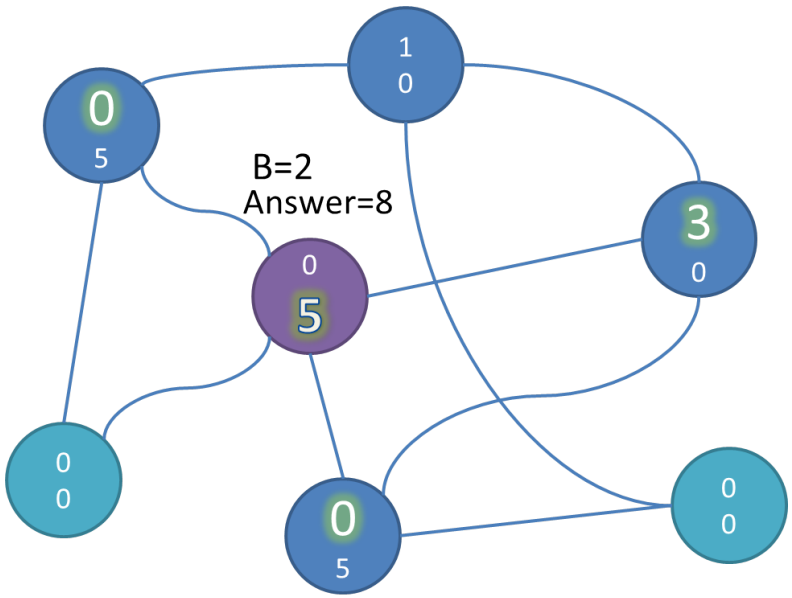
$Query(x)$ details

- $\sum v[heavy\ \ neighbors] = \sum_{y\ \ is\ \ heavy\ \ neighbor} vh[y]$
- $\sum v[light\ \ neighbors] = sl[x]$
- $O(1 + cnt\_heavy)$ time
- $cnt\_heavy \cdot B \leq 2m = O(n)$



$Query(x)$ details

- $\sum v[heavy\ \ neighbors] = \sum_{y\ \ is\ \ heavy\ \ neighbor} vh[y]$
- $\sum v[light\ \ neighbors] = sl[x]$
- $O(1 + cnt\_heavy)$ time
- $cnt\_heavy \cdot B \leq 2m = O(n)$
- $O(n/B)$ time



Balance

- $O(n)$ space

Balance

- $O(n)$ space
- $O(B)$ time for add

Balance

- $O(n)$ space
- $O(B)$ time for add
- $O(n/B)$ time for query

Balance

- $O(n)$ space
- $O(B)$ time for add
- $O(n/B)$ time for query
- make $B = \sqrt{n}$

Balance

- $O(n)$ space
- $O(B)$ time for add
- $O(n/B)$ time for query
- make $B = \sqrt{n}$
- $O(\sqrt{n})$ for each operation