1. 1. Lists are vertical

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|  | 20 |  |  | 16 | 44 | 94 | 12 |  | 13 |  |
|  |  |  |  | 5 | 88 | 39 | 23 |  |  |  |
|  |  |  |  |  | 11 |  |  |  |  |  |



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 11 | 39 | 20 | 5 | 16 | 44 | 88 | 12 | 23 | 13 | 94 |



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 20 |  | 16 | 11 | 39 | 44 | 88 | 12 | 23 | 13 | 94 |

* 1. Assuming the mod11 is applied last

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 11 | 23 | 20 | 16 | 44 | 5 | 94 | 12 | 88 | 13 | 39 |



put(k,v):

i <- h(k)

j <- 0

while A[i] != NULL or A[i] = SpecialMarker do

if A[i].key = k then

A[i] <- (k,v) //replace the old value with same key

j <- j+1

i <- h(k) + j\*j

A[i] <- (k,v) //found a nice empty index

get(k):

i <- h(k)

j <- 0

while A[i] != NULL do

if (A[i].key = k) then

return A[i]

j <- j+1

i <- h(k) +j\*j

return NULL //not found

remove(k):

i <- h(k)

j <- 0

while A[i] != NULL do

if (A[i].key = k) then

temp <- A[i]

A[i] <- SpecialMarker //a non-NULL key that will never be used by regular values to represent a removed element

return temp

j <- j+1

i <- h(k) + j\*j

return NULL //key not found



**Algorithm** matrixDFS(𝐺,𝑣,P)

**Input**: A graph 𝐺,with 𝑛 vertices labeled 0,...,𝑛−1,represented as an adjacency matrix, a starting vertex 𝑣, and an array C of size n that is initially 0s

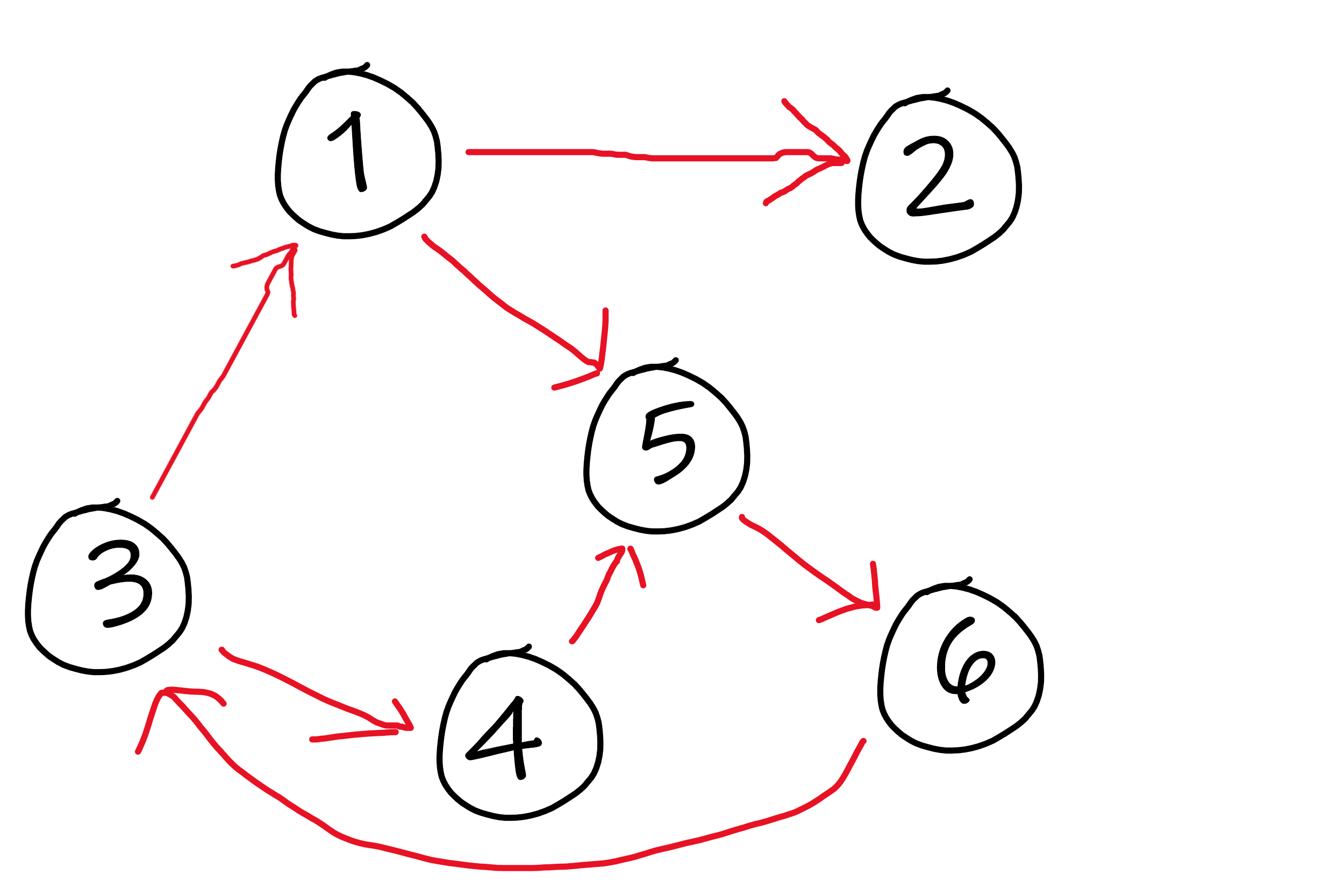
**Output**: An integer array of size 𝑛, containing the preorder labelling of the vertices.

1. P[0] <- v
2. C[v] <- 1
3. for i from 0 to n do
4. if G[v][i] is not 0 then
5. if C[i] = 1 then skip
6. Else append DFS(G,i,C) to P
7. return P
9. Takes 1 unit of time to append
10. Takes 1 unit of time
11. Loops n times per iteration
12. Takes 1 time per loop
13. Takes 1 unit of time to check
14. Recursive call
15. Takes 1 unit of time

This algorithm is O(n2) in the best and worst case, because it will always conduct n loops over n iterations.

The adjacency list representation has a best/worst case of O(|V| + |E|), where |V| is the number of vertices in the graph and |E| is the number of edges in the graph.





G0 =

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | 0 | 1 | 0 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 1 | 0 | 0 | 1 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 1 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 1 |
| **6** | 0 | 0 | 1 | 0 | 0 | 0 |

Rows are from, columns are to.



G1 =

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | 0 | 1 | 0 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 1 | 1 | 0 | 1 | 1 | 0 |
| **4** | 0 | 0 | 0 | 0 | 1 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 1 |
| **6** | 0 | 0 | 1 | 0 | 0 | 0 |

G2 has no changes since 2 has no outgoing edges

G3 =

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | 0 | 1 | 0 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 1 | 1 | 0 | 1 | 1 | 0 |
| **4** | 0 | 0 | 0 | 0 | 1 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 1 |
| **6** | 1 | 1 | 1 | 1 | 1 | 0 |

G4 has no changes since its only incoming edges, (3,4) and now (6,4), already have outgoing edges towards 5.

G5 =

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | 0 | 1 | 0 | 0 | 1 | 1 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 1 | 1 | 0 | 1 | 1 | 1 |
| **4** | 0 | 0 | 0 | 0 | 1 | 1 |
| **5** | 0 | 0 | 0 | 0 | 0 | 1 |
| **6** | 1 | 1 | 1 | 1 | 1 | 0 |

G6 =

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | 0 | 1 | 1 | 1 | 1 | 1 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 1 | 1 | 0 | 1 | 1 | 1 |
| **4** | 1 | 1 | 1 | 0 | 1 | 1 |
| **5** | 1 | 1 | 1 | 1 | 0 | 1 |
| **6** | 1 | 1 | 1 | 1 | 1 | 0 |