CS3000: Algorithms & Data — Summer '23 — Laney Strange

BFS Algorithm

BFS(G, s)

```
 1  for each vertex in u ∈ G.V − {s}
 2       u.color = WHITE
 3       u.d = ∞
 4       u.π = NIL
 5  s.color = GRAY
 6  s.d = 0
 7  s.π = NIL
 8  Q = ∅
 9  ENQUEUE(Q, s)
10  while Q ≠ ∅
11       u = DEQUEUE(Q)
12       for each v ∈ G.Adj[u]
13            if v.color == WHITE
14                 v.color = GRAY
15                 v.d = u.d + 1
16                 v.π = u
17                 ENQUEUE(Q, v)
18       u.color = BLACK
```

Here's how we typeset the above functions in CLRS style:

```
\begin{codebox}
\Procname{$\proc{BFS}(G, s)$}
\li \For each vertex in $u \in G.V - \{s\}$
\Do
\li $\id{u.color} \gets \const{White}$
\li $u.d \gets \infty$
\li $u.\pi \gets \const{Nil}$
\End
\li $\id{s.color} \gets \const{Gray}$
\li $s.d \gets 0$
\li $s.\pi \gets \const{Nil}$
\li $Q \gets \emptyset$
\li $\proc{Enqueue}(Q, s)$
\li \While $Q \ne \emptyset$
\Do
\li $u \gets \proc{Dequeue}(Q)$
\li \For each $v \in G.Adj[u]$
\Do
```

```
\li \If $\id{v.color} == \const{White}$
\Then
\li $\id{v.color} \gets \const{Gray}$
\li $v.d \gets u.d + 1$
\li $v.\pi \gets u$
\li $\proc{Enqueue}(Q, v)$
\End
\li $\id{u.color} \gets \const{Black}$
\end{codebox}
```