

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 class BinaryTree {
7 public:
8     int value;
9     BinaryTree *left = NULL;
10    BinaryTree *right = NULL;
11
12    BinaryTree(int value);
13 };
14
15 void mutate(BinaryTree *node, BinaryTree *parent, bool isLeftChild);
16
17 // O(n) time | O(d) space - where n is the number of nodes in the Bina
18 // and d is the depth (height) of the Binary Tree
19 BinaryTree *rightSiblingTree(BinaryTree *root) {
20     mutate(root, NULL, false);
21     return root;
22 }
23
24 void mutate(BinaryTree *node, BinaryTree *parent, bool isLeftChild) {
25     if (node == NULL)
26         return;
27
28     auto left = node->left;
29     auto right = node->right;
30     mutate(left, node, true);
31     if (parent == NULL) {
32         node->right = NULL;
33     } else if (isLeftChild) {
```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 // This is the class of the input root. Do not edit it.
5 class BinaryTree {
6 public:
7     int value;
8     BinaryTree *left = NULL;
9     BinaryTree *right = NULL;
10
11    BinaryTree(int value);
12 };
13
14 BinaryTree *rightSiblingTree(BinaryTree *root) {
15     // Write your code here.
16     return root;
17 }
18
```

Our Tests

Custom Output

Submit Code

```
1 // Example 1:
2 // Input: root = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
3 // Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
4 // Explanation: The root node has a left child 2 and a right child 3. Node 2 has a left child 4 and a right child 5. Node 3 has a left child 6 and a right child 7. Node 4 has a left child 8 and a right child 9. Node 5 has a left child 10 and a right child NULL.
```

```

1  class PageRanker:
2      def __init__(self, graph):
3          self.graph = graph
4          self.pagerank = {}
5          self.damping = 0.85
6          self.max_iter = 100
7          self.tolerance = 1e-6
8
9      def get_neighbors(self, node):
10         return self.graph.get(node, [])
11
12     def calculate_pagerank(self):
13         # Initialize pagerank values
14         num_nodes = len(self.graph)
15         for node in self.graph:
16             self.pagerank[node] = 1.0 / num_nodes
17
18         # Iterate until convergence
19         for i in range(self.max_iter):
20             new_pagerank = {}
21             for node in self.graph:
22                 in_links = self.get_in_links(node)
23                 total_in = sum(pagerank[src] for src in in_links)
24                 new_pagerank[node] = (1 - self.damping) / num_nodes + \
25                     self.damping * total_in
26
27             # Check for convergence
28             diff = max(abs(new_pagerank[node] - self.pagerank[node])
29                        for node in self.graph)
30             if diff < self.tolerance:
31                 break
32             self.pagerank = new_pagerank
33
34     def get_in_links(self, node):
35         in_links = []
36         for src, targets in self.graph.items():
37             if node in targets:
38                 in_links.append(src)
39         return in_links
40
41     def get_pagerank(self, node):
42         return self.pagerank.get(node, 0)
43
44     def __str__(self):
45         return str(self.pagerank)
46
47     def __repr__(self):
48         return f"PageRanker({self.graph})"
49
50     def __eq__(self, other):
51         return isinstance(other, PageRanker) and \
52             self.graph == other.graph
53
54     def __hash__(self):
55         return hash(self.graph)
56
57     def __len__(self):
58         return len(self.pagerank)
59
60     def __getitem__(self, node):
61         return self.pagerank.get(node, 0)
62
63     def __setitem__(self, node, value):
64         self.pagerank[node] = value
65
66     def __delitem__(self, node):
67         del self.pagerank[node]
68
69     def __iter__(self):
70         return iter(self.pagerank)
71
72     def __reversed__(self):
73         return reversed(self.pagerank)
74
75     def __contains__(self, node):
76         return node in self.pagerank
77
78     def __bool__(self):
79         return bool(self.pagerank)
80
81     def __nonzero__(self):
82         return bool(self.pagerank)
83
84     def __call__(self, node):
85         return self.get_pagerank(node)
86
87     def __getattr__(self, attr):
88         if attr in self.__dict__:
89             return self.__dict__[attr]
90         else:
91             raise AttributeError(f"PageRanker object has no attribute {attr}")
92
93     def __setattr__(self, attr, value):
94         self.__dict__[attr] = value
95
96     def __delattr__(self, attr):
97         del self.__dict__[attr]
98
99     def __copy__(self):
100        return PageRanker(self.graph)
101
102     def __deepcopy__(self, memo):
103        return PageRanker(self.graph)
104
105     def __reduce__(self):
106        return (PageRanker, (self.graph,))
107
108     def __getstate__(self):
109        return self.graph
110
111     def __setstate__(self, state):
112        self.graph = state
113
114     def __pickle__(self):
115        return self.__reduce__()
116
117     def __unpickle__(self, state):
118        self.__setstate__(state)
119
120     def __str__(self):
121        return str(self.pagerank)
122
123     def __repr__(self):
124        return f"PageRanker({self.graph})"
125
126     def __eq__(self, other):
127        return isinstance(other, PageRanker) and \
128            self.graph == other.graph
129
130     def __hash__(self):
131        return hash(self.graph)
132
133     def __len__(self):
134        return len(self.pagerank)
135
136     def __getitem__(self, node):
137        return self.pagerank.get(node, 0)
138
139     def __setitem__(self, node, value):
140        self.pagerank[node] = value
141
142     def __delitem__(self, node):
143        del self.pagerank[node]
144
145     def __iter__(self):
146        return iter(self.pagerank)
147
148     def __reversed__(self):
149        return reversed(self.pagerank)
150
151     def __contains__(self, node):
152        return node in self.pagerank
153
154     def __bool__(self):
155        return bool(self.pagerank)
156
157     def __nonzero__(self):
158        return bool(self.pagerank)
159
160     def __call__(self, node):
161        return self.get_pagerank(node)
162
163     def __getattr__(self, attr):
164        if attr in self.__dict__:
165            return self.__dict__[attr]
166        else:
167            raise AttributeError(f"PageRanker object has no attribute {attr}")
168
169     def __setattr__(self, attr, value):
170        self.__dict__[attr] = value
171
172     def __delattr__(self, attr):
173        del self.__dict__[attr]
174
175     def __copy__(self):
176        return PageRanker(self.graph)
177
178     def __deepcopy__(self, memo):
179        return PageRanker(self.graph)
180
181     def __reduce__(self):
182        return (PageRanker, (self.graph,))
183
184     def __getstate__(self):
185        return self.graph
186
187     def __setstate__(self, state):
188        self.graph = state
189
190     def __pickle__(self):
191        return self.__reduce__()
192
193     def __unpickle__(self, state):
194        self.__setstate__(state)
195
196     def __str__(self):
197        return str(self.pagerank)
198
199     def __repr__(self):
200        return f"PageRanker({self.graph})"
201
202     def __eq__(self, other):
203        return isinstance(other, PageRanker) and \
204            self.graph == other.graph
205
206     def __hash__(self):
207        return hash(self.graph)
208
209     def __len__(self):
210        return len(self.pagerank)
211
212     def __getitem__(self, node):
213        return self.pagerank.get(node, 0)
214
215     def __setitem__(self, node, value):
216        self.pagerank[node] = value
217
218     def __delitem__(self, node):
219        del self.pagerank[node]
220
221     def __iter__(self):
222        return iter(self.pagerank)
223
224     def __reversed__(self):
225        return reversed(self.pagerank)
226
227     def __contains__(self, node):
228        return node in self.pagerank
229
230     def __bool__(self):
231        return bool(self.pagerank)
232
233     def __nonzero__(self):
234        return bool(self.pagerank)
235
236     def __call__(self, node):
237        return self.get_pagerank(node)
238
239     def __getattr__(self, attr):
240        if attr in self.__dict__:
241            return self.__dict__[attr]
242        else:
243            raise AttributeError(f"PageRanker object has no attribute {attr}")
244
245     def __setattr__(self, attr, value):
246        self.__dict__[attr] = value
247
248     def __delattr__(self, attr):
249        del self.__dict__[attr]
250
251     def __copy__(self):
252        return PageRanker(self.graph)
253
254     def __deepcopy__(self, memo):
255        return PageRanker(self.graph)
256
257     def __reduce__(self):
258        return (PageRanker, (self.graph,))
259
260     def __getstate__(self):
261        return self.graph
262
263     def __setstate__(self, state):
264        self.graph = state
265
266     def __pickle__(self):
267        return self.__reduce__()
268
269     def __unpickle__(self, state):
270        self.__setstate__(state)
271
272     def __str__(self):
273        return str(self.pagerank)
274
275     def __repr__(self):
276        return f"PageRanker({self.graph})"
277
278     def __eq__(self, other):
279        return isinstance(other, PageRanker) and \
280            self.graph == other.graph
281
282     def __hash__(self):
283        return hash(self.graph)
284
285     def __len__(self):
286        return len(self.pagerank)
287
288     def __getitem__(self, node):
289        return self.pagerank.get(node, 0)
290
291     def __setitem__(self, node, value):
292        self.pagerank[node] = value
293
294     def __delitem__(self, node):
295        del self.pagerank[node]
296
297     def __iter__(self):
298        return iter(self.pagerank)
299
300     def __reversed__(self):
301        return reversed(self.pagerank)
302
303     def __contains__(self, node):
304        return node in self.pagerank
305
306     def __bool__(self):
307        return bool(self.pagerank)
308
309     def __nonzero__(self):
310        return bool(self.pagerank)
311
312     def __call__(self, node):
313        return self.get_pagerank(node)
314
315     def __getattr__(self, attr):
316        if attr in self.__dict__:
317            return self.__dict__[attr]
318        else:
319            raise AttributeError(f"PageRanker object has no attribute {attr}")
320
321     def __setattr__(self, attr, value):
322        self.__dict__[attr] = value
323
324     def __delattr__(self, attr):
325        del self.__dict__[attr]
326
327     def __copy__(self):
328        return PageRanker(self.graph)
329
330     def __deepcopy__(self, memo):
331        return PageRanker(self.graph)
332
333     def __reduce__(self):
334        return (PageRanker, (self.graph,))
335
336     def __getstate__(self):
337        return self.graph
338
339     def __setstate__(self, state):
340        self.graph = state
341
342     def __pickle__(self):
343        return self.__reduce__()
344
345     def __unpickle__(self, state):
346        self.__setstate__(state)
347
348     def __str__(self):
349        return str(self.pagerank)
350
351     def __repr__(self):
352        return f"PageRanker({self.graph})"
353
354     def __eq__(self, other):
355        return isinstance(other, PageRanker) and \
356            self.graph == other.graph
357
358     def __hash__(self):
359        return hash(self.graph)
360
361     def __len__(self):
362        return len(self.pagerank)
363
364     def __getitem__(self, node):
365        return self.pagerank.get(node, 0)
366
367     def __setitem__(self, node, value):
368        self.pagerank[node] = value
369
370     def __delitem__(self, node):
371        del self.pagerank[node]
372
373     def __iter__(self):
374        return iter(self.pagerank)
375
376     def __reversed__(self):
377        return reversed(self.pagerank)
378
379     def __contains__(self, node):
380        return node in self.pagerank
381
382     def __bool__(self):
383        return bool(self.pagerank)
384
385     def __nonzero__(self):
386        return bool(self.pagerank)
387
388     def __call__(self, node):
389        return self.get_pagerank(node)
390
391     def __getattr__(self, attr):
392        if attr in self.__dict__:
393            return self.__dict__[attr]
394        else:
395            raise AttributeError(f"PageRanker object has no attribute {attr}")
396
397     def __setattr__(self, attr, value):
398        self.__dict__[attr] = value
399
400     def __delattr__(self, attr):
401        del self.__dict__[attr]
402
403     def __copy__(self):
404        return PageRanker(self.graph)
405
406     def __deepcopy__(self, memo):
407        return PageRanker(self.graph)
408
409     def __reduce__(self):
410        return (PageRanker, (self.graph,))
411
412     def __getstate__(self):
413        return self.graph
414
415     def __setstate__(self, state):
416        self.graph = state
417
418     def __pickle__(self):
419        return self.__reduce__()
420
421     def __unpickle__(self, state):
422        self.__setstate__(state)
423
424     def __str__(self):
425        return str(self.pagerank)
426
427     def __repr__(self):
428        return f"PageRanker({self.graph})"
429
430     def __eq__(self, other):
431        return isinstance(other, PageRanker) and \
432            self.graph == other.graph
433
434     def __hash__(self):
435        return hash(self.graph)
436
437     def __len__(self):
438        return len(self.pagerank)
439
440     def __getitem__(self, node):
441        return self.pagerank.get(node, 0)
442
443     def __setitem__(self, node, value):
444        self.pagerank[node] = value
445
446     def __delitem__(self, node):
447        del self.pagerank[node]
448
449     def __iter__(self):
450        return iter(self.pagerank)
451
452     def __reversed__(self):
453        return reversed(self.pagerank)
454
455     def __contains__(self, node):
456        return node in self.pagerank
457
458     def __bool__(self):
459        return bool(self.pagerank)
460
461     def __nonzero__(self):
462        return bool(self.pagerank)
463
464     def __call__(self, node):
465        return self.get_pagerank(node)
466
467     def __getattr__(self, attr):
468        if attr in self.__dict__:
469            return self.__dict__[attr]
470        else:
471            raise AttributeError(f"PageRanker object has no attribute {attr}")
472
473     def __setattr__(self, attr, value):
474        self.__dict__[attr] = value
475
476     def __delattr__(self, attr):
477        del self.__dict__[attr]
478
479     def __copy__(self):
480        return PageRanker(self.graph)
481
482     def __deepcopy__(self, memo):
483        return PageRanker(self.graph)
484
485     def __reduce__(self):
486        return (PageRanker, (self.graph,))
487
488     def __getstate__(self):
489        return self.graph
490
491     def __setstate__(self, state):
492        self.graph = state
493
494     def __pickle__(self):
495        return self.__reduce__()
496
497     def __unpickle__(self, state):
498        self.__setstate__(state)
499
500     def __str__(self):
501        return str(self.pagerank)
502
503     def __repr__(self):
504        return f"PageRanker({self.graph})"
505
506     def __eq__(self, other):
507        return isinstance(other, PageRanker) and \
508            self.graph == other.graph
509
510     def __hash__(self):
511        return hash(self.graph)
512
513     def __len__(self):
514        return len(self.pagerank)
515
516     def __getitem__(self, node):
517        return self.pagerank.get(node, 0)
518
519     def __setitem__(self, node, value):
520        self.pagerank[node] = value
521
522     def __delitem__(self, node):
523        del self.pagerank[node]
524
525     def __iter__(self):
526        return iter(self.pagerank)
527
528     def __reversed__(self):
529        return reversed(self.pagerank)
530
531     def __contains__(self, node):
532        return node in self.pagerank
533
534     def __bool__(self):
535        return bool(self.pagerank)
536
537     def __nonzero__(self):
538        return bool(self.pagerank)
539
540     def __call__(self, node):
541        return self.get_pagerank(node)
542
543     def __getattr__(self, attr):
544        if attr in self.__dict__:
545            return self.__dict__[attr]
546        else:
547            raise AttributeError(f"PageRanker object has no attribute {attr}")
548
549     def __setattr__(self, attr, value):
550        self.__dict__[attr] = value
551
552     def __delattr__(self, attr):
553        del self.__dict__[attr]
554
555     def __copy__(self):
556        return PageRanker(self.graph)
557
558     def __deepcopy__(self, memo):
559        return PageRanker(self.graph)
560
561     def __reduce__(self):
562        return (PageRanker, (self.graph,))
563
564     def __getstate__(self):
565        return self.graph
566
567     def __setstate__(self, state):
568        self.graph = state
569
570     def __pickle__(self):
571        return self.__reduce__()
572
573     def __unpickle__(self, state):
574        self.__setstate__(state)
575
576     def __str__(self):
577        return str(self.pagerank)
578
579     def __repr__(self):
580        return f"PageRanker({self.graph})"
581
582     def __eq__(self, other):
583        return isinstance(other, PageRanker) and \
584            self.graph == other.graph
585
586     def __hash__(self):
587        return hash(self.graph)
588
589     def __len__(self):
590        return len(self.pagerank)
591
592     def __getitem__(self, node):
593        return self.pagerank.get(node, 0)
594
595     def __setitem__(self, node, value):
596        self.pagerank[node] = value
597
598     def __delitem__(self, node):
599        del self.pagerank[node]
600
601     def __iter__(self):
602        return iter(self.pagerank)
603
604     def __reversed__(self):
605        return reversed(self.pagerank)
606
607     def __contains__(self, node):
608        return node in self.pagerank
609
610     def __bool__(self):
611        return bool(self.pagerank)
612
613     def __nonzero__(self):
614        return bool(self.pagerank)
615
616     def __call__(self, node):
617        return self.get_pagerank(node)
618
619     def __getattr__(self, attr):
620        if attr in self.__dict__:
621            return self.__dict__[attr]
622        else:
623            raise AttributeError(f"PageRanker object has no attribute {attr}")
624
625     def __setattr__(self, attr, value):
626        self.__dict__[attr] = value
627
628     def __delattr__(self, attr):
629        del self.__dict__[attr]
630
631     def __copy__(self):
632        return PageRanker(self.graph)
633
634     def __deepcopy__(self, memo):
635        return PageRanker(self.graph)
636
637     def __reduce__(self):
638        return (PageRanker, (self.graph,))
639
640     def __getstate__(self):
641        return self.graph
642
643     def __setstate__(self, state):
644        self.graph = state
645
646     def __pickle__(self):
647        return self.__reduce__()
648
649     def __unpickle__(self, state):
650        self.__setstate__(state)
651
652     def __str__(self):
653        return str(self.pagerank)
654
655     def __repr__(self):
656        return f"PageRanker({self.graph})"
657
658     def __eq__(self, other):
659        return isinstance(other, PageRanker) and \
660            self.graph == other.graph
661
662     def __hash__(self):
663        return hash(self.graph)
664
665     def __len__(self):
666        return len(self.pagerank)
667
668     def __getitem__(self, node):
669        return self.pagerank.get(node, 0)
670
671     def __setitem__(self, node, value):
672        self.pagerank[node] = value
673
674     def __delitem__(self, node):
675        del self.pagerank[node]
676
677     def __iter__(self):
678        return iter(self.pagerank)
679
680     def __reversed__(self):
681        return reversed(self.pagerank)
682
683     def __contains__(self, node):
684        return node in self.pagerank
685
686     def __bool__(self):
687        return bool(self.pagerank)
688
689     def __nonzero__(self):
690        return bool(self.pagerank)
691
692     def __call__(self, node):
693        return self.get_pagerank(node)
694
695     def __getattr__(self, attr):
696        if attr in self.__dict__:
697            return self.__dict__[attr]
698        else:
699            raise AttributeError(f"PageRanker object has no attribute {attr}")
700
701     def __setattr__(self, attr, value):
702        self.__dict__[attr] = value
703
704     def __delattr__(self, attr):
705        del self.__dict__[attr]
706
707     def __copy__(self):
708        return PageRanker(self.graph)
709
710     def __deepcopy__(self, memo):
711        return PageRanker(self.graph)
712
713     def __reduce__(self):
714        return (PageRanker, (self.graph,))
715
716     def __getstate__(self):
717        return self.graph
718
719     def __setstate__(self, state):
720        self.graph = state
721
722     def __pickle__(self):
723        return self.__reduce__()
724
725     def __unpickle__(self, state):
726        self.__setstate__(state)
727
728     def __str__(self):
729        return str(self.pagerank)
730
731     def __repr__(self):
732        return f"PageRanker({self.graph})"
733
734     def __eq__(self, other):
735        return isinstance(other, PageRanker) and \
736            self.graph == other.graph
737
738     def __hash__(self):
739        return hash(self.graph)
740
741     def __len__(self):
742        return len(self.pagerank)
743
744     def __getitem__(self, node):
745        return self.pagerank.get(node, 0)
746
747     def __setitem__(self, node, value):
748        self.pagerank[node] = value
749
750     def __delitem__(self, node):
751        del self.pagerank[node]
752
753     def __iter__(self):
754        return iter(self.pagerank)
755
756     def __reversed__(self):
757        return reversed(self.pagerank)
758
759     def __contains__(self, node):
760        return node in self.pagerank
761
762     def __bool__(self):
763        return bool(self.pagerank)
764
765     def __nonzero__(self):
766        return bool(self.pagerank)
767
768     def __call__(self, node):
769        return self.get_pagerank(node)
770
771     def __getattr__(self, attr):
772        if attr in self.__dict__:
773            return self.__dict__[attr]
774        else:
775            raise AttributeError(f"PageRanker object has no attribute {attr}")
776
777     def __setattr__(self, attr, value):
778        self.__dict__[attr] = value
779
780     def __delattr__(self, attr):
781        del self.__dict__[attr]
782
783     def __copy__(self):
784        return PageRanker(self.graph)
785
786     def __deepcopy__(self, memo):
787        return PageRanker(self.graph)
788
789     def __reduce__(self):
790        return (PageRanker, (self.graph,))
791
792     def __getstate__(self):
793        return self.graph
794
795     def __setstate__(self, state):
796        self.graph = state
797
798     def __pickle__(self):
799        return self.__reduce__()
800
801     def __unpickle__(self, state):
802        self.__setstate__(state)
803
804     def __str__(self):
805        return str(self.pagerank)
806
807     def __repr__(self):
808        return f"PageRanker({self.graph})"
809
810     def __eq__(self, other):
811        return isinstance(other, PageRanker) and \
812            self.graph == other.graph
813
814     def __hash__(self):
815        return hash(self.graph)
816
817     def __len__(self):
818        return len(self.pagerank)
819
820     def __getitem__(self, node):
821        return self.pagerank.get(node, 0)
822
823     def __setitem__(self, node, value):
824        self.pagerank[node] = value
825
826     def __delitem__(self, node):
827        del self.pagerank[node]
828
829     def __iter__(self):
830        return iter(self.pagerank)
831
832     def __reversed__(self):
833        return reversed(self.pagerank)
834
835     def __contains__(self, node):
836        return node in self.pagerank
837
838     def __bool__(self):
839        return bool(self.pagerank)
840
841     def __nonzero__(self):
842        return bool(self.pagerank)
843
844     def __call__(self, node):
845        return self.get_pagerank(node)
846
847     def __getattr__(self, attr):
848        if attr in self.__dict__:
849            return self.__dict__[attr]
850        else:
851            raise AttributeError(f"PageRanker object has no attribute {attr}")
852
853     def __setattr__(self, attr, value):
854        self.__dict__[attr] = value
855
856     def __delattr__(self, attr):
857        del self.__dict__[attr]
858
859     def __copy__(self):
860        return PageRanker(self.graph)
861
862     def __deepcopy__(self, memo):
863        return PageRanker(self.graph)
864
865     def __reduce__(self):
866        return (PageRanker, (self.graph,))
867
868     def __getstate__(self):
869        return self.graph
870
871     def __setstate__(self, state):
872        self.graph = state
873
874     def __pickle__(self):
875        return self.__reduce__()
876
877     def __unpickle__(self, state):
878        self.__setstate__(state)
879
880     def __str__(self):
881        return str(self.pagerank)
882
883     def __repr__(self):
884        return f"PageRanker({self.graph})"
885
886     def __eq__(self, other):
887        return isinstance(other, PageRanker) and \
888            self.graph == other.graph
889
890     def __hash__(self):
891        return hash(self.graph)
892
893     def __len__(self):
894        return len(self.pagerank)
895
896     def __getitem__(self, node):
897        return self.pagerank.get(node, 0)
898
899     def __setitem__(self, node, value):
900        self.pagerank[node] = value
901
902     def __delitem__(self, node):
903        del self.pagerank[node]
904
905     def __iter__(self):
906        return iter(self.pagerank)
907
908     def __reversed__(self):
909        return reversed(self.pagerank)
910
911     def __contains__(self, node):
912        return node in self.pagerank
913
914     def __bool__(self):
915        return bool(self.pagerank)
916
917     def __nonzero__(self):
918        return bool(self.pagerank)
919
920     def __call__(self, node):
921        return self.get_pagerank(node)
922
923     def __getattr__(self, attr):
924        if attr in self.__dict__:
925            return self.__dict__[attr]
926        else:
927            raise AttributeError(f"PageRanker object has no attribute {attr}")
928
929     def __setattr__(self, attr, value):
930        self.__dict__[attr] = value
931
932     def __delattr__(self, attr):
933        del self.__dict__[attr]
934
935     def __copy__(self):
936        return PageRanker(self.graph)
937
938     def __deepcopy__(self, memo):
939        return PageRanker(self.graph)
940
941     def __reduce__(self):
942        return (PageRanker, (self.graph,))
943
944     def __getstate__(self):
945        return self.graph
946
947     def __setstate__(self, state):
948        self.graph = state
949
950     def __pickle__(self):
951        return self.__reduce__()
952
953     def __unpickle__(self, state):
954        self.__setstate__(state)
955
956     def __str__(self):
957        return str(self.pagerank)
958
959     def __repr__(self):
960        return f"PageRanker({self.graph})"
961
962     def __eq__(self, other):
963        return isinstance(other, PageRanker) and \
964            self.graph == other.graph
965
966     def __hash__(self):
967        return hash(self.graph)
968
969     def __len__(self):
970        return len(self.pagerank)
971
972     def __getitem__(self, node):
973        return self.pagerank.get(node, 0)
974
975     def __setitem__(self, node, value):
976        self.pagerank[node] = value
977
978     def __delitem__(self, node):
979        del self.pagerank[node]
980
981     def __iter__(self):
982        return iter(self.pagerank)
983
984     def __reversed__(self):
985        return reversed(self.pagerank)
986
987     def __contains__(self, node):
988        return node in self.pagerank
989
990     def __bool__(self):
991        return bool(self.pagerank)
992
993     def __nonzero__(self):
994        return bool(self.pagerank)
995
996     def __call__(self, node):
997        return self.get_pagerank(node)
998
999     def __getattr__(self, attr):
1000       if attr in self.__dict__:
1001           return self.__dict__[attr]
1002       else:
1003           raise AttributeError(f"PageRanker object has no attribute {attr}")
1004
1005     def __setattr__(self, attr, value):
1006        self.__dict__[attr] = value
1007
1008     def __delattr__(self, attr):
1009        del self.__dict__[attr]
1010
1011     def __copy__(self):
1012        return PageRanker(self.graph)
1013
1014     def __deepcopy__(self, memo):
1015        return PageRanker(self.graph)
1016
1017     def __reduce__(self):
1018        return (PageRanker, (self.graph,))
1019
1020     def __getstate__(self):
1021        return self.graph
1022
1023     def __setstate__(self, state):
1024        self.graph = state
1025
1026     def __pickle__(self):
1027        return self.__reduce__()
1028
1029     def __unpickle__(self, state):
1030        self.__setstate__(state)
1031
1032     def __str__(self):
1033        return str(self.pagerank)
1034
1035     def __repr__(self):
1036        return f"PageRanker({self.graph})"
1037
1038     def __eq__(self, other):
1039        return isinstance(other, PageRanker) and \
1040            self.graph == other.graph
1041
1042     def __hash__(self):
1043        return hash(self.graph)
1044
1045     def __len__(self):
1046        return len(self.pagerank)
1047
1048     def __getitem__(self, node):
1049        return self.pagerank.get(node, 0)
1050
1051     def __setitem__(self, node, value):
1052        self.pagerank[node] = value
1053
1054     def __delitem__(self, node):
1055        del self.pagerank[node]
1056
1057     def __iter__(self):
1058        return iter(self.pagerank)
1059
1060     def __reversed__(self):
1061        return reversed(self.pagerank)
1062
1063     def __contains__(self, node):
1064        return node in self.pagerank
1065
1066     def __bool__(self):
1067        return bool(self.pagerank)
1068
1069     def __nonzero__(self):
1070        return bool(self.pagerank)
1071
1072     def __call__(self, node):
1073        return self.get_pagerank(node)
1074
1075     def __getattr__(self, attr):
1076        if attr in self.__dict__:
1077            return self.__dict__[attr]
1078        else:
1079            raise AttributeError(f"PageRanker object has no attribute {attr}")
1080
1081     def __setattr__(self, attr, value):
1082        self.__dict__[attr] = value
1083
1084     def __delattr__(self, attr):
1085        del self.__dict__[attr]
1086
1087     def __copy__(self):
1088        return PageRanker(self.graph)
1089
1090     def __deepcopy__(self, memo):
1091        return PageRanker(self.graph)
1092
1093     def __reduce__(self):
1094        return (PageRanker, (self.graph,))
1095
1096     def __getstate__(self):
1097        return self.graph
1098
1099     def __setstate__(self, state):
1100        self.graph = state
1101
1102     def __pickle__(self):
1103        return self.__reduce__()
1104
1105     def __unpickle__(self, state):
1106        self.__setstate__(state)
1107
1108     def __str__(self):
1109        return str(self.pagerank)
1110
1111     def __repr__(self):
1112        return f"PageRanker({self.graph})"
1113
1114     def __eq__(self, other):
1115        return isinstance(other, PageRanker) and \
1116            self.graph == other.graph
1117
1118     def __hash__(self):
1119        return hash(self.graph)
1120
1121     def __len__(self):
1122        return len(self.pagerank)
1123
1124     def __getitem__(self, node):
1125        return self.pagerank.get(node, 0)
1126
1127     def __setitem__(self, node, value):
1128        self.pagerank[node] = value
1129
1130     def __delitem__(self, node):
1131        del self.pagerank[node]
1132
1133     def __iter__(self):
1134        return iter(self.pagerank)
1135
1136     def __reversed__(self):
1137        return reversed(self.pagerank)
1138
1139     def __contains__(self, node):
1140        return node in self.pagerank
1141
1142     def __bool__(self):
1143        return bool(self.pagerank)
1144
1145     def __nonzero__(self):
1146        return bool(self.pagerank)
1147
1148     def __call__(self, node):
1149        return self.get_pagerank(node)
1150
1151     def __getattr__(self, attr):
1152        if attr in self.__dict__:
1153            return self.__dict__[attr]
1154        else:
1155            raise AttributeError(f"PageRanker object has no attribute {attr}")
1156
1157     def __setattr__(self, attr, value):
1158        self.__dict__[attr] = value
1159
1160     def __delattr__(self, attr):
1161        del self.__dict__[attr]
1162
1163     def __copy__(self):
1164        return PageRanker(self.graph)
1165
1166     def __deepcopy__(self, memo):
1167        return PageRanker(self.graph)
1168
1169     def __reduce__(self):
1170        return (PageRanker, (self.graph,))
1171
1172     def __getstate__(self):
1173        return self.graph
1174
1175     def __setstate__(self, state):
1176        self.graph = state
1177
1178     def __pickle__(self):
1179        return self.__reduce__()
1180
1181     def __unpickle__(self, state):
1182        self.__setstate__(state)
1183
1184     def __str__(self):
1185        return str(self.pagerank)
1186
1187     def __repr__(self):
1188        return f"PageRanker({self.graph})"
1189
1190     def __eq__(self, other):
1191        return isinstance(other, PageRanker) and \
1192            self.graph == other.graph
1193
1194     def __hash__(self):
1195        return hash(self.graph)
1196
1197     def __len__(self):
1198        return len(self.pagerank)
1199
1200     def __getitem__(self, node):
1201        return self.pagerank.get(node, 0)
1202
1203     def __setitem__(self, node, value):
1204        self.pagerank[node] = value
1205
1206     def __delitem__(self, node):
1207        del self.pagerank[node]
1208
1209     def __iter__(self):
1210        return iter(self.pagerank)
1211
1212     def __reversed__(self):
1213        return reversed(self.pagerank)
1214
1215     def __contains__(self, node):
1216        return node in self.pagerank
1217
1218     def __bool__(self):
1219        return bool(self.pagerank)
1220
1221     def __nonzero__(self):
1222        return bool(self.pagerank)
1223
1224     def __call__(self, node):
1225        return self.get_pagerank(node)
1226
1227     def __getattr__(self, attr):
1228        if attr in self.__dict__:
1229            return self.__dict__[attr]
1230        else:
1231            raise AttributeError(f"PageRanker object has no attribute {attr}")
1232
1233     def __setattr__(self, attr, value):
1234        self.__dict__[attr] = value
1235
1236     def __delattr__(self, attr):
1237        del self.__dict__[attr]
1238
1239     def __copy__(self):
1240        return PageRanker(self.graph)
1241
1242     def __deepcopy__(self, memo):
1243        return PageRanker(self.graph)
1244
1245     def __reduce__(self):
1246        return (PageRanker, (self.graph,))
1247

```