

۱ توصیف الگوریتم $BFS(s)$

این الگوریتم از راس s گراف ساده $G = (V, E)$ شروع کرده و همه رئوسی که از s قابل دسترسی هستند به نوبت علامت می‌زنند. ترتیب علامت خوردن بستگی به فاصله از s دارد. ابتدا راسهایی که مستقیماً به s وصل هستند علامت می‌خورند (لایه اول)، سپس راسهایی که قبلاً علامت نخوردند و به لایه اول متصل هستند علامت می‌خورند (لایه دوم)، و به همین ترتیب لایه‌های بعدی شکل می‌گیرند.

کار الگوریتم را می‌توان به یک سری مرحله تقسیم کرد. در شروع مرحله i ام، لیست $L[i]$ موجود است که در مرحله قبلی ساخته شده است. در این مرحله راسهای لایه $i + 1$ ام شناسایی شده و به لیست پیوندی $L[i + 1]$ اضافه می‌شوند. در واقع در شروع هر مرحله یک لیست جدید (برای نگهداری راسهای لایه جدید) ساخته می‌شود. اگر در مرحله i ام لیست $L[i]$ تهی باشد (وقتی اتفاق می‌افتد که در مرحله قبلی هیچ راسی علامت نخورده باشد) الگوریتم پایان می‌پذیرد.

از آرایه منطقی D برای ثبت علامت راسها استفاده می‌کنیم به این معنی که $D[u] = \text{true}$ اگر و فقط اگر راس u علامت خورده باشد است. فرض بر این است که گراف G بصورت لیست مجاورتی به الگوریتم داده شده است. الگوریتم در حین پیمایش گراف درخت BFS را نیز می‌سازد.

۱. $D[s] \leftarrow \text{true}$

۲. $\forall v \in V/\{s\} \ D[v] \leftarrow \text{false}$

۳. $L[0]$ را ایجاد کن. سپس $(s) \leftarrow L[0]$ (لایه صفرم تنها شامل راس s است)

۴. $i \leftarrow 0$ (شمارنده مرحله)

۵. $T \leftarrow \emptyset$ (درخت BFS در آغاز کار تهی است)

۶. تا زمانی که $L[i]$ تهی نیست اعمال زیر را انجام بده:

۱.۶ لیست جدید $L[i + 1]$ را ایجاد کن. در شروع کار لیست خالی است.

۲.۶ برای هر راس u در لیست $L[i]$ اعمال زیر را انجام بده:

– هر بار یک یال (u, v) که روی راس u قرار دارد را در نظر بگیر.

– اگر $D[v] = \text{false}$ آنگاه:

$D[v] \leftarrow \text{true}$

یال (u, v) را به درخت T اضافه کن.

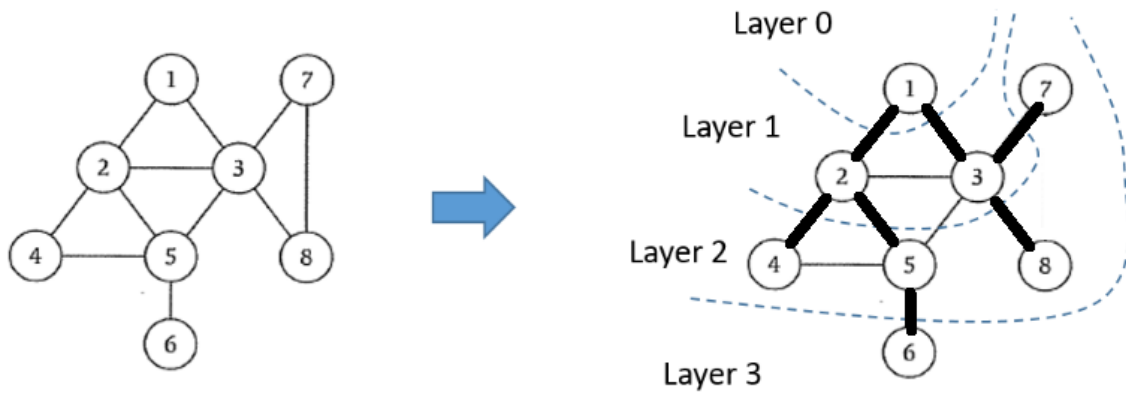
راس v را به لیست $L[i + 1]$ اضافه کن.

۳.۶ $i \leftarrow i + 1$

۷. پایان

۲ یک مثال

در مثال الگوریتم BFS را از راس شماره 1 شروع کرده‌ایم. لایه‌ها در شکل نشان داده شده‌اند. درخت BFS با خطوط پررنگ مشخص شده است.



$$L[0] = (1) \quad L[1] = (2, 3) \quad L[2] = (4, 5, 7, 8) \quad L[3] = (6) \quad L[4] = \emptyset$$