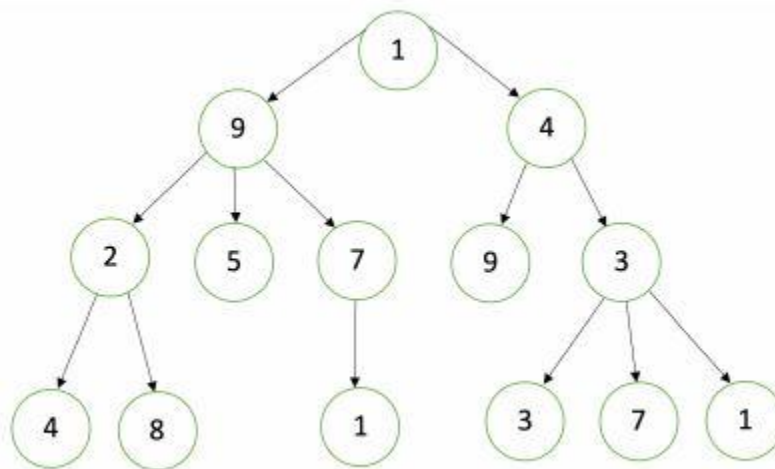


روی درخت DFS میزنیم و مسیر رسیدن به هر راس را هم به صورت یک زوج مرتب نگه میداریم. یعنی برای مثال وقتی به راس 2 در این مثال میرسیم $(2, \{1, 9\})$ را خواهیم داشت. به هر راسی که می رسیم آن را با مسیری که در اختیار داریم چک میکنیم که آیا حضور دارد یا نه. اگر حضور نداشت این راس را به مجموعه راس های طلایی اضافه میکنیم و اگر حضور داشت دیگر به بررسی فرزندانش نمی پردازیم و سراغ آخرین راسی که به رنگ خاکستری بوده میرویم (طبق الگوریتم DFS راس هایی که به عنوان فرزند راس های قبلی آورده شده اند ولی هنوز سراغ فرزندان آن ها نرفته ایم خاکستری هستند، راس هایی که پیموده شده و سراغ فرزندانشان هم رفتیم سیاه هستند و راس های سفید هنوز دیده نشده اند). مثلاً پیاده کردن این الگوریتم در درخت سوال با فرض این که وقتی اولویت ها برابر هستند سراغ راس چپ تر می رویم:



ابتدا $(1, \{\})$ دیده میشود و لذا 1 به جواب ها اضافه می شود. سپس به ترتیب 9 و 2 و 4 و 8 و 5 و 7 به راس های طلایی اضافه می شوند اما وقتی به $(1, \{1, 9, 7\})$ میرسیم میبینیم که در مسیر تکرار وجود دارد و دیگر ادامه نمی دهیم و سراغ آخرین راس خاکستری یعنی 4 (بالا سمت راست) می رویم. که مسلماً $(4, \{1\})$ تکراری وجود ندارد و طلایی است. باز به همین شکل 9 و 3 اضافه می شوند ولی 3 پایینی نمیشود 7 میشود و 1 آخر هم نمی شود. و جواب خواهد بود: $[1, 9, 2, 4, 8, 5, 7, 4, 9, 3, 7]$