

خلاصه کاری که این الگوریتم DF انجام می دهد:

در الگوریتم DF از آیم حالت شروع را در نظر می گیریم بعد یک پیشه تعریف می کنیم و آن حالت شروع را به پیشه اضافه می کنیم و یک آرایه برای آن های که دیده شده اند در هر مرحله در نظر

می گیریم بعد زمانی که پیشه صافی نیست می آیم آن تیره و آن مسیر را از بالای پیشه بر می داریم و اگر آن تیره یک هدف باشد آن مسیر را بر می داریم، اگر هم نبود آن را به دیده شده ها (پیمایش شده اند) اضافه می کنیم و آن مسیرهای که می توانیم حرکت کنیم با آن تیره را در Successor قرار می دهیم. بعد در حلقه به ازای آن action و state و Cost در آن Successor می آیم اگر اون حالت (state) هنوز دیده نشده است آن حالت و آن مسیر به علاوه ی آن [direction] را به پیشه اضافه می کنیم.

خلاصه کاری که این الگوریتم BFS انجام می دهد:

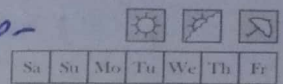
در الگوریتم BFS از آیم حالت شروع را در نظر می گیریم و بعد یک صف تعریف می کنیم و آن حالت شروع را به صف اضافه می کنیم و یک آرایه برای آن های که دیده شده اند در هر مرحله در نظر می گیریم و بعد آن حالت شروع را به دیده شده ها اضافه می کنیم. بعد زمانی که صف صافی نیست می آیم آن تیره و آن

مسیر را از بالای پیشه بر می داریم و اگر آن تیره یک هدف باشد آن مسیر را بر می داریم، بعد آن مسیرهای که می توانیم برویم با آن تیره را در Successor قرار می دهیم. بعد در حلقه به ازای آن action و state و Cost در آن Successor می آیم اگر آن حالت هنوز مشاهده نشده، آن حالت و آن مسیر به علاوه ی آن [direction] را به صف اضافه می کنیم و بعد آن حالت را به دیده شده ها اضافه می کنیم.

خلاصه کاری که این الگوریتم UCS انجام می دهد:

در الگوریتم UCS از آیم از یک صف اولویت استفاده می کنیم و بعد یک آرایه برای دیده شده ها در نظر می گیریم و یک آرایه برای مسیرهای در نظر می گیریم و مجموع هزینه را با total cost را با برابری با صفر قرار می دهیم و بعد حالت شروع را به دست می آوریم و بعد آن حالت شروع و آن هزینه اش را به آن صف اولویت اضافه می کنیم.

Subject: مسئله صف اولویت می آید بر اساس آن اولویت شان آن گره ها را انتخاب می کنند و به ما
 Year: _____ Month: _____ Day: _____



بعد از زمانی که صف اولویت ما خالی نیست می آییم آن گره و آن مسیر را از صف اولویت می برداریم و
 آن گره یک هدف باشد آن مسیر را بر می گردانیم و بعد یک حلقه می گذاریم برای آن successor
 direction و cost در آن گره که می آید آن را Successor (مسیری که به ما می دهد)
 تا الان دیده نشده (پیمایش نشده است) آن را به دیده شده ها اضافه کن و بعد گره جدید را در نظر می گیریم که
 Successor و آن مسیر به علاوه آن [direction] هست و بعد آن گره جدید و آن هزینه اش
 را در صف اولویت قرار می دهیم و بعد می گوییم آن را Successor یک هدف است؛ گره جدید را در نظر
 می گیریم و بعد آن گره جدید و آن هزینه اش را در صف اولویت قرار می دهیم.
 - اگر هم نتوانستیم مسیری پیدا کنیم return None را بر می گردانیم.

خدا صه کاری که این الگوریتم A^* انجام می دهد:

در الگوریتم A^* مثل همان الگوریتم UCS است با این تفاوت که مجموعه هزینه را با
 total cost جایگزین می دهیم که می آید بر اساس آن heuristic که خود مسئله متغیر ترفقه
 به آن heuristic آن حالت شروع را می دهیم و آن را در این total cost نگه می داریم برای هر
 حالت.

و بعد هم داده همان الگوریتم UCS است و بعد برای آن گره های که می خواهیم به
 صف اولویت اضافه کنیم آن heuristic هم حساب می کنیم و به آن هزینه اضافه می کنیم و به آن
 گره نسبت می دهیم و به صف اولویت اضافه می کنیم.