به نام خدا

محمدمهدی هجرتی	**۹۷۲۳۱۰۰**
پروژه ی اول هوش	**۳ دی ۹۹**

فرموله سازی مسئله

حالات ممکن: هر چیدمان ممکن از کارت ها در ردیف ها

حالت اولیه: یکی از حالات اولیه به صورت رندوم

اعمال: جا به جایی پایین ترین کارت هر لیست و قرار دادن آن در یک ردیف دیگر در صورتی که عدد آن از کارتی که روی آن قرار می گیرد کمتر باشد.

آزمون هدف: یکسان بودن و نزولی بودن کارت های تمام لیست ها

نحوه ی پیاده سازی

برای حل مسئله یک کلاس نود در نظر گرفته شد که در خود والد و فرزندانش را نگه داری میکند. و با یک لینکد لیست از هر ردیف کارت نگه داری می شود. هر کارت یک آبجکت از کلاس کارت است که در آن رنگ و شماره ی آن ذخیره شده است.

سوال اول الگوریتم BFS

در کلاس BFS فرآیند حل با این روش انجام میشود. حل با روش جست و جوی گرافی انجام شده است. پس یک لیست برای نود ها explored شده وجود دارد. که هر بار که نود تولید میشود در صورتی که در این لیست موجود باشد، آن نود نادیده گرفته میشود. برای نود های تولید شده یک لیست FIFO به نام frontier در نظر گرفته شده است که هر بار نود تولیدی به آن اضافه میشود و از سمت دیگر برای بررسی خارج می شود. در ضمن هر بار بررسی نود در هنگام بسط نود انجام میشود. یعنی پس از براشتن از صف frontier بررسی انجام می شود.

سوال دوم الگوریتم IDS

کلیت این الگوریتم بر مبنای DFS پیاده سازی می شود. DFS مشابه BFS است با این تفاوت که صف frontier از نوع LIFO می باشد و به همین دلیل هر بار ابتدا عمیق ترین نود چک میشود. حال به اجرای

DFS تا یک عمق مشخص DLS گفته میشود. الگوریتم IDS در یک حلقه هر بار DLS را تا عمق های متفاوت چک میکند. در این صورت از مزایای هر دو الگوریتم BFS و DFS یعنی حافظه ی کمتر و کامل بودن بهره مند میشویم.

سوال سوم الگوریتم *A

کلیت این الگوریتم بر مبنای UCS نوشته میشود با این تفاوت که در اینجا یک تابع هیوریستیک برای کاهش چک کردن حالت های پرت به کمک ما می آید. الگوریتم UCS خود بر مبنای BFS است با این تفاوت که هر بار کم هزینه ترین نود را برای بسط انتخاب کرده و مهم تر از آن اگر نودی تولید شد و با حالت یکسان هزینه ی کمتری از نود دیگر در frontier داشت با آن جایگزین میشود.

تابع هیوریستیک استفاده شده

برای هیوریستیک این مسئله از ترکیب و در واقع ماکسیمم دو حالت زیر استفاده شده است.

۱. تعداد رنگ های متفاوت: به این صورت که در هر کدام از لیست ها از پایین تا بالا کارت های روی هم چک میشوند و در صورت تفاوت رنگ آن ها یک واحد به هیوریستیک اضافه می شود.
۲. تعداد اعداد غیر نزولی: از پایین تا بالا هر لیست بررسی شده و در صورتی که از عدد قبلی بیشتر باشد یک واحد اضافه میشود.

برای هر لیست ماکسیمم این عدد محاسبه و اعمال میشود.

قابل قبول است چون برای هر عدد اضافه شده به آن حدقل یک اکشن لازم است انجام شود.

مقایسه ی روش ها

گره های بسط داده شده (پیچیدگی فضایی):

با توجه به گرافی بودن جست و جوی BFS و بررسی نود ها در هنگام بسط، حافظه ی بیشتری لازم دارد.

و IDS با توجه به اینکه مبتنی بر DFS است، از همه کمتر به حافظه نیاز دارد.

گره های بسط داده شده (پیچیدگی فضایی):

BFS به حافظه ی بیشتری لازم دارد. با توجه به اینکه برای تولید فرزندان به والد نیاز است و هر بار سطی جلو میرویم.

و IDS با توجه به اینکه مبتنی بر DFS است، از همه کمتر به حافظه نیاز دارد.

گره های گره های تولید شده (پیچیدگی زمانی):

BFS با توجه به گرافی بودن جست و جو و اینکه در هنگام بسط بررسی شرط را انجام میدهد بیشتر است.

در حالت کلی به صورت زیر می باشد.

با توجه به اینکه !(m + n + k) = b حالت می تواند داشته باشد.

	گره های بسط داده شده (پیچیدگی فضایی)	گره های تولید شده (پیچیدگی زمانی)	عمق جواب
BFS	$O(b^d)$	$O(b^{d+1})$	d
IDS	$O(bd)$	$O(b^d)$	d
A*	نمایی	نمایی	d