اثبات قابل قبول بودن تابع ابتکاری (heuristic function admissibility proof):

```python
def get_h_n(self):
    h = 0
    colors = {'red': 0, 'blue': 0, 'green': 0, 'yellow': 0}
    for pipe in self.pipes:
        for ball in pipe.stack:
            colors[ball] += 1
    number_of_balls = sum(colors.values())
    for color in colors:
        n = colors[color] / self.pipes[0].limit
        for pipe in self.pipes:
            if n == 0 and color == pipe.color:
                pipe.color = None
            if color == pipe.color:
                n -= 1
    for pipe in self.pipes:
        if pipe.color is not None:
            for ball in range(len(pipe.stack)):
                if pipe.stack[ball] != pipe.color:
                    h += len(pipe.stack) - ball
        elif pipe.color is None:
            h += len(pipe.stack)
    return h
```

پیاده سازی این تابع در کد پروژه به این صورت است:

حال در این پیاده سازی یک پراپرتی ب نام color برای هر لوله در نظر گرفته شده است که نشان می دهد بیشترین رنگی که در این لوله است چیست

حال نحوه عملکرد این تابع به گونه ای است که در ابتدا لوله هایی که قرار نیست با توپ های همرنگ پر شوند را بدون رنگ در نظر می گیرد:

pipe.color = None

حال با یک الگوریتم محاسبه می کنیم توپ هایی که در لوله های غیر همرنگ با خود اند در ساده ترین حالت در چند حرکت می توانند از لوله غیرهمرنگی که در آن است خارج شود.

حالت اول این است که لوله مورد بررسی رنگی داشته باشد پس مهم است که توپ غیر همرنگ در کجای این لوله است:

If pipe.color is not None: . . .

h += len(pipe.stack) – ball(ایندکس آن توپ)

در حالت بالا برای مثال اگر توپی قرمز در لوله ای با رنگ آبی قرار دارد و دو توپ آبی بالای آن اند حال میخواهد از این لوله خارج شود پس نیاز به سه حرکت دارد، دو حرکت برای خروج دو توپ آبی و یک حرکت برای خروج خود پس : h += 3

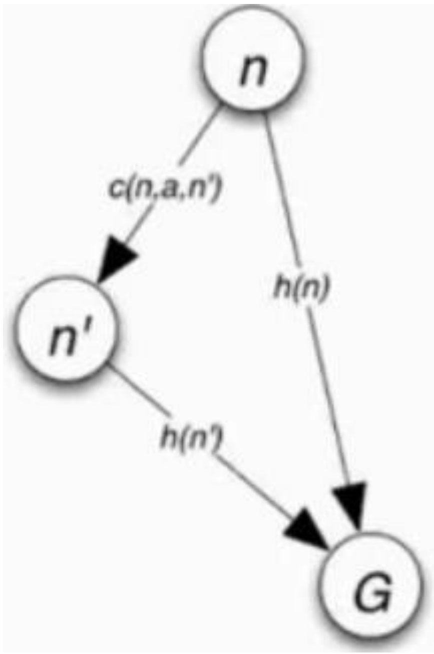
حالت دوم این است که لوله مورد بررسی رنگی ندارد پس در نهایت این لوله باید خالی شود پس ما به تعداد توپ های این لوله برای خالی کردن آن به حرکت نیاز داریم :

elif pipe.color is None:

h += len(pipe.stack)

این شرحی از روند الگوریتم بود که نشان می دهد حالات آنقدر ساده فرض شده اند که در بیشترین حالت h برابر h* خواهد بود زیرا ما حداقل تعداد حرکت های ضروری برای خالی کردن یا همرنگ کردن توپ های لوله ها را محاسبه می کنیم که در بیشترین حالت h* نیز به آن حرکت ها نیاز خواهد داشت تا بتواند به ادامه حرکت های خود بپردازد پس تخمین بیش از حد نخواهیم زد پس : همیشه h <= h*

پس قابل قبول بودن تابع h ثابت شد.



اثبات مستقل بودن تابع ابتکاری (proof heuristic function consistency):

در ابتدا فرض می کنیم تابع h ما مستقل نیست پس یعنی $h(n')+c(n,a,n')< h(n)$

که به صورت بدیهی نتیجه میشود: $f(n')<f(n)$

حال باید به خلاف این فرض برسیم تا ثابت کنیم تابع h ما مستقل است

در ابتدا به این اشاره میکنیم که تابع g(n) را طوری طراحی کرده ایم که همیشه اگر از گره ای به گره دیگر برویم که آن گره مقصد هدف نیست هزینه ما مثبت خواهد بود یعنی c مقدار مثبتی خواهد داشتو در ضمن g(n) طوری طراحی شده است که اگر گره n دو فرزند n' و G (که هدف است یا نزدیک به آن است)داشته باشد هزینه رفتن به گره G کمتر یا مساوی هزینه رفتن به گره n' است پس اگر هزینه واقعی رفتن از n به G برابر d باشد نتیجه می شود که $g(n)+d<= g(n) + c + h(n')$

حال از آنجا که h قاابل قبول است پس $h(n) <= d$ پس با در نظر گرفتن دو این دو نامساوی به این نامساوی میرسیم که خلاف چیزی است که فرض کرده ایم.

$$g(n) + d <= g(n) + c + h(n') \ , \ h(n) <= d \Rightarrow g(n) + h(n) <= g(n) + c + h(n') \Rightarrow h(n) <= h(n') + c$$