

## در مسیر بهشت

در یکی از جهان های موازی به علت محدودیت منابع غذایی، گونه های جانوری در خطر انقراض اند. گفته می شود در بخشی از جهان بهشتی وجود دارد که منابع غذایی نامحدودی دارد وظیفه ما این است که این گونه ها را به سمت بهشت راهنمایی کنیم. در طول بازی شما به دلخواه کنترل یکی از گونه ها را برعهده خواهید داشت و گونه های دیگر باید توسط برنامه ای که شما برایشان می نویسید راهنمایی شوند.

### جهان (World):

جهان بازی در واقع یک نقشه مربعی شکل به ضلع  $m$  است ( $m$  در ورودی داده می شود) که جانداران و منابع غذایی و ... در هر یک از خانه های موجود در این نقشه جای می گیرند.

هر خانه از نقشه شامل یکی از چهار حالت زیر است:

- یک جاندار
- یک منبع غذایی با انرژی مشخص
- بن بست (حیوانات نمی توانند به این خانه وارد شوند).
- خانه خالی (جانداران می توانند به این خانه ها وارد شوند)

### بهشت (Heaven):

بهشت یک زیر مجموعه از جهان است که در آن غذای بی نهایت وجود دارد و هر گونه ای که بتواند به این منطقه برسد از خطر انقراض نجات پیدا کرده و بازی خاتمه می یابد.

## حیوانات (Animals):

حیوانات، موجوداتی هستند ما سعی داریم آنها را از خطر انقراض نجات دهیم. هر حیوان یک سری ویژگی هایی دارد که در ژن آنها ذخیره می شود.

ژن یک حیوان شامل موارد زیر است:

۱. میزان انرژی برای جابجا شدن به اندازه ی یک خانه
۲. تعداد خانه هایی که می تواند در یک حرکت جابجا شود (حداقل یک و حداکثر سه )
۳. حداقل انرژی مورد نیاز برای تولید مثل
۴. قدرت حمله
۵. قدرت دفاع

حیوانات با غذا خوردن می توانند انرژی لازم برای انجام کار های خود را به دست آورند.

## غذا (Food):

هر غذا یک انرژی مشخص دارد که بعد از مصرف به میزان انرژی حیوان مصرف کننده اضافه خواهد شد.

## بن بست (Dead End):

حیوانات نمی توانند از روی خانه های بن بست عبور کنند.

## فاز صفرم:

در این فاز برای آشنایی بیشتر با فرآیند پیاده سازی پروژه، باید برنامه ای بنویسید که ابعاد جهان، مختصات خانه های بن بست، خانه های بهشت و همینطور مختصات اولیه حیوانات را دریافت کند و صرف نظر از اینکه آیا این حیوانات انرژی کافی برای رسیدن به بهشت در طول مسیر دارند، آنها را به سمت بهشت راهنمایی کنیم. برای مثال ورودی برنامه به شکل زیر خواهد بود:

```
20
# 5 (5,5)(6,6)(10,10)(0,1)(19,19)
H 3 (1,1)(1,0)(2,1)
===
F 100 (3,2)
F 150 (2,4)
F 200 (1,5)
F 300 (0,19)
F 120 (0,5)
===
A 2 (1,4)(2,3)
B 2 (1,3)(0,3)
===
A
```

## توضیحات:

در خط اول یک عدد که ابعاد جهان ( $m$ ) را مشخص می کند داده شده است.  
در خط دوم که با # شروع می شود تعداد و مکان خانه های بن بست آمده است.  
که در آن برای مثال عدد ۵ به معنای آن است که در این نقشه ۵ بن بست داریم و بعد از آن با یک فاصله زوج مرتب هایی که نشان دهنده مکان این پنج بن بست است می آید که مولفه اول نشان دهنده سطر و مولفه دوم

نشان دهند ستون مورد نظر در نقشه است.

در خط بعدی که با H شروع میشود و بعد از آن تعداد خانه هایی که بهشت هستند و با یک فاصله بعد زوج مرتب هایی که نشان دهنده مکان این خانه هاست می آید.

خط بعد شامل سه کاراکتر === خواهد بود و بعد از آن در هر خط که با F شروع می شود عدد بعد از آن نمایانگر انرژی آن غذا و زوج مرتب بعد از آن نشان دهنده مکان آن غذا است.

در خط بعد از آخرین غذا سه کاراکتر === خواهد بود و بعد از آن در هر خط که با یک حرف انگلیسی (غیر از F و H) است شروع خواهد شد نشان دهنده حیوانات است. در هر کدام از این خطوط، عدد اول نشان دهنده تعداد از آن گونه و زوج مرتب های بعدی نشان دهنده مکان هر کدام از حیوانات آن گونه است.

در پایان سه کاراکتر === خواهد آمد. بعد از آن گونه ای که می خواهید کنترل کنید وارد می شود.

خانه هایی که در نقشه خالی هستند با کاراکتر «.» و بقیه خانه ها با اولین کاراکتری برای هر مورد در اول خطوط آمده است مشخص خواهند شد و هر خانه با خانه های کناری خود به اندازه یک کاراکتر (space)، فاصله دارد. در تمامی فاز های بازی باید در هر دور و بعد از تصمیم گیری همه جانوران، حرکت هایی که آن ها در دور فعلی انجام داده اند را چاپ کنید و یک تا سه ثانیه صبر کنید و دور بعد را انجام دهد. برای زیبایی می توان در هنگام چاپ نقشه و اطلاعات دور جدید موارد قبلی موجود در ترمینال را پاک کرد.

دقت کنید که حیوانات یک گونه را شما کنترل می کنید و باید مکانیزیمی برای انتخاب حرکت برای آنها پیاده سازی کنید و بقیه گونه ها توسط برنامه شما کنترل می شود.

برای نمونه نقشه چاپی برای یک جهان ۵ در ۵ به شکل زیر خواهد بود:

A . # . .

A . B . .

# . B . .

. . # . .

H . . H .

## فاز اول:

در این بخش یک فایل حاوی نقشه ، مکان اولیه حیوانات و ژن های آنها و مکان غذا ها و انرژی هر کدام به نام map\_phase1.txt در اختیار شما قرار می گیرد. شما در فاز باید این فایل را خوانده و براساس محتوای این فایل اقدام به حرکت دادن حیوانات در نقشه به سمت بهشت کنید. در این فاز هم نیازی به در نظر گرفتن انرژی و مصرف غذا نیست اما حیوان می تواند بر اساس تعداد خانه هایی ممکن برای جابجایی در یک حرکت که در ژن او آماده است انتخاب کند. برای مثال اگر این مقدار برابر دو باشد یک حیوان در این فاز چون تنها انتخاب حرکت کردن را دارد می تواند یک یا دو خانه در هر هشت جهت مجاور خود حرکت کند به شرط آنکه در این مسیر بن بست وجود نداشته باشد.

-محتویات نمونه فایل map:

```
20
# 5 (5,5)(6,6)(10,10)(0,1)(19,19)
H 3 (1,1)(1,0)(2,1)
===
F 100 (3,2)
F 150 (2,4)
F 200 (1,5)
F 300 (0,19)
F 120 (0,5)
===
A 2 (1,4)(2,3)
B 2 (1,3)(0,3)
===
A
===
A 300 50$1$350$100$300
B 250 25$2$500$200$40
```

## توضیحات:

موارد ابتدایی مانند فاز صفرم هستند و تنها در این بخش در پایان فایل، ژن ها قرار می گیرند.

در خط بعد از گونه انتخاب شده برای کاربر سه کاراکتر === خواهد بود و بعد از آن در هر خط که ابتدا با یک نام حیوان شروع می شود ژن حیوانات را خواهیم داشت. اولین عدد نشان دهنده انرژی اولیه حیوانات در ابتدای بازی است و بعد از آن پنج عدد صحیح که با علامت \$ از هم جدا می شوند آمده است که در واقع رشته ژن حیوانات است و اعداد به ترتیب از چپ به راست نشانگر

- «میزان انرژی برای جابجا شدن به اندازه ی یک خانه»
- «تعداد خانه هایی که می تواند در یک حرکت جابجا شود»
- «انرژی مورد نیاز برای تولید مثل»
- «قدرت حمله»
- «قدرت دفاع»

است.

در هر فاز نقشه مناسب آن فاز به شما داده خواهد شد که نام آن به فرمت

map\_phasen.txt

است که n شماره فاز خواهد بود.

## فاز دوم:

در این فاز هر حیوان در طی مسیر به سمت بهشت با توجه به ویژگی های خود که در ژن آن قرار دارد برای هر حرکت انرژی ای مصرف خواهد کرد. و اگر در طی یکی از پیمایش های حلقه بازی انرژی کافی برای انجام هیچ حرکتی نداشته باشد، خواهد مرد و باقی مانده انرژی او به شکل یک غذا در خانه فعلی او باقی خواهد ماند.

پس ممکن است در طی حرکت نیاز باشد که یک حیوان غذا مصرف کند. هر حیوانی که داخل یکی از خانه های حاوی غذا وارد شود کل انرژی آن غذا به انرژی فعلی حیوان اضافه خواهد شد.

## فاز سوم:

در این فاز می خواهیم تولید مثل را برای گونه ها پیاده سازی کنیم. هرگاه دو جاندار از یک گونه در خانه های مجاور یکدیگر باشند و سطح انرژی لازم برای تولید مثل را داشته باشند می توانند یک موجود جدید ایجاد کنند و در این صورت به میزان نصف انرژی مورد نیاز برای تولید مثل از میزان انرژی فعلی آنها کسر خواهد شد. فرزند متولد شده شامل انرژی اولیه ذکر شده در فایل نقشه است و در نزدیکترین خانه خالی به والدین خود قرار خواهد گرفت اما هر کدام از ویژگی های ژن او به شکل تصادفی می تواند زیاد یا کم باشد. دقت کنید که در صورتی که دو جاندار در کنار هم قرار بگیرند و شرایط فوق را داشته باشند الزامی به تولید مثل نیست. برای جهش ژنتیکی محدودیت های زیر را در نظر داشته باشید:

- «میزان انرژی برای جابجا شدن به اندازه ی یک خانه» کمترین: ۲۰ ، بیشترین: ۲۰۰
- «تعداد خانه هایی که می تواند در یک حرکت جابجا شود» کمترین: ۱، بیشترین: ۳
- «انرژی مورد نیاز برای تولید مثل» کمترین: ۶۰، بیشترین ۶۰۰
- «قدرت حمله» کمترین ۲۰، بیشترین ۵۰۰
- «قدرت دفاع» کمترین ۲۰، بیشترین ۵۰۰



## فاز چهارم:

در این فاز می خواهیم جنگیدن بین گونه ها و همکاری بین همنوعان را پیاده سازی کنیم.

در هر یک از پیمایش های بازی هر جاندار می تواند به جای حرکت کردن انرژی لازم برای حرکت یک خانه خود را به یکی از همنوعان خود ببخشد.

اگر دو جاندار از گونه های مختلف در خانه های مجاور هم باشند یک از آنها می تواند به دیگری اعلام جنگ کند.

اگر قدرت تهاجمی گونه مهاجم بیشتر از قدرت دفاع گونه دیگر باشد برنده جنگ است و گونه مدافع خواهد مرد و انرژی فعلی او به شکل غذا به گونه پیروز خواهد رسید و در غیر اینصورت گونه مهاجم خواهد مرد و اتفاق مشابه برای اون خواهد افتاد. دقت کنید که هزینه اعلام جنگ برای گونه مهاجم برابر انرژی مصرف شده برای حرکت به اندازه سه خانه خواهد بود. بعد از جنگ مکان فعلی گونه پیروز حفظ خواهد شد.

از آنجایی که جانداران میزان انرژی، قدرت دفاعی و تهاجمی و بقیه مشخصات دیگران را می بینند می توانند از این موارد برای تصمیم گیری استفاده کنند.

## موارد نمره اضافه:

- تصمیم گیری هوشمند در هر لحظه با استفاده از الگوریتم های هوش مصنوعی (برای مثال کوتاه ترین مسیر ممکن با در نظر گرفتن انرژی فعلی و غذا های موجود در طول مسیر)
- پیاده سازی یک تولید کننده نقشه که فایلی مشابه نقشه هر فاز را تولید کند.
- تمام حرکات و وضعیت نقشه را در یک فایل ذخیره کنید به صورتی که human-readable باشد. (لاگ کردن)
- برنامه ای بنویسید که با ورودی دادن لاگ یک بازی آن بازی را از نو اجرا کند (در اینجا نیازی به تصمیم گیری نیست چون قبلا تصمیمات گرفته شده و آن ها را دوباره تکرار می کند.) دقت کنید که توانید برای سادگی کار یک لاگ مجزا برای این امر داشته باشید که تنها در آن به صورت کد شده حرکت ها و وضعیت ها را نگه میدارید.
- برنامه بخش قبل خود را به گونه ای تغییر دهید که بتوان بازی ای که از میانه آن قطع شده است را ادامه داد

## نکات تکمیلی:

- ❖ پروژه به صورت گروهی و در قالب گروه های ۳ نفره انجام خواهد شد.
- ❖ در طی انجام پروژه باید از گیت (Git) برای کنترل ورژن و کار گروهی استفاده کنید.
- ❖ تمام افراد گروه باید به پروژه تسلط داشته باشند.
- ❖ در صورت مشاهده هرگونه تقلب برای همه افراد خاطی نمره ۱۰۰- در نظر گرفته خواهد شد.

## با آرزوی موفقیت

تیم حل تمرین مبانی کامپیوتر و برنامه نویسی – پاییز ۱۴۰۰