

الگوریتم ژنتیک :

سوال ۱ :

کروموزومی که تعریف شده یک آرایه تشکیل شده از ۴۱ ضریب است. و این ضرایب در جمعیت اولیه به صورت رندوم و بین اعداد ۱۰- و ۱۰ ایجاد می‌شوند که اگر بخواهیم اعداد را حقیقی فرض کنیم که فضای حالت بینهایت می‌شود چون اعداد در یک رنج پیوسته قرار می‌گیرند که بینهایت حالت داریم. ولی اگر فرضاً فضا را محدود کنیم اما اگر اعداد صحیح را در نظر بگیریم. 20^{41} حالت داریم.

سوال ۲ :

یکی از راه‌ها این است که موقع انتخاب کردن، همیشه تعداد n تا از جمعیتان که بیشترین ریت را دارند انتخاب کنیم یا حداقل شانس انتخابشان را افزایش دهیم.
راه بعدی پیدا کردن میوتیشن ریت متناسب و افزایش آن اگر تغییر زیادی نمی‌بینیم یا کاهش اگر تغییرات باعث افزایش randomness بین ژن‌ها می‌شود.

سوال ۳ :

انتخاب بهترین‌ها : این روش شاید ساده ترین روش و اولین چیزی که به ذهن می‌آید باشد که ممکن است نتیجه خوبی داشته باشد اما ممکن است باعث شود که فقط یک سری ژن تکراری انتخاب شوند و عملاً پیشرفتی رخ ندهد.

انتخاب براساس رنک : اینجا هم ایده برداشتن بهترین‌هاست ولی مطلق نیست و با رنک کردن داده‌ها همچنان داده‌های بهتر شانس انتخاب شدن بیشتری دارند ولی داده‌های دیگر هم انتخاب می‌شوند و این به diversity و احتمالاً پیشرفت سریعتر الگوریتم کمک می‌کند.

سوال ۴ :

راه اول افزایش mutation rate است که چون اساساً mutation برای ساخت ژن‌های جدید است و شاید ساخت ژن‌هایی که در جمعیت اولیه وجود ندارند پس با افزایش نرخ آن احتمال بوجود آوردن ژن‌های جدید بیشتر است.

راه دوم می‌توانیم به ژن‌هایی که متفاوت اند امتیاز بیشتری دهیم و مثلاً در سیستم انتخاب بر اساس رنک ژن‌هایی که متفاوت اند رنک بالاتری بگیرند تا شانس انتخابشان بیشتر شود و در نتیجه ژن‌های جدید تری تولید شوند.

سوال ۵ :

فرمول :

$$R^2 = 1 - (\text{sum}((y_i - \hat{y}_i)^2) / \text{sum}((y_i - \bar{y})^2))$$

این فرمول برای محاسبه دقت مدل در تخمین است و در آن y می‌شود داده اصلی و \hat{y} می‌شود داده تخمین زده شده \bar{y} هم میانگین است. مقدار خروجی بین ۰ و ۱ است و ۰ به معنی این است که مدل دیتا را در حد میانگین تخمین زده و ۱ یعنی مدل دقیقاً داده‌ها را تخمین زده است.

الگوریتم minimax

سوال ۱ :

هرچه عمق بیشتر باشد ما حرکت‌های بیشتری از حریف را می‌توانیم ببینیم و انتخاب‌های بهتری کنیم پس احتمال برد بیشتر می‌شود اما از نظر زمان با افزایش عمق زمان لازم بیشتر می‌شود چون گره‌های بیشتری را برای هر حرکت می‌بینیم.

سوال ۲ :

بله - می‌توانیم این کار را بکنیم برای مثال یکی از راه حل‌ها move ordering است که برای مکسیمی‌ز کردن قبل از انتخاب حرکت حرکتی که احتمال دارد امتیاز بیشتری بدهد را انتخاب می‌کنیم برای مثال در بازی پنتاگو انتخاب موقعیت‌های وسط از لحاظ استراتژیک احتمال افزایش امتیاز بیشتری دارد پس آن‌ها انتخاب می‌شوند.

سوال ۳ :

برنچ فکتورینگ به تعداد حرکت‌های در دسترس در هر مرحله از درخت بازی گفته می‌شود که برای بازی ما اول ۳۶ خانه خالی و ۸ حالت چرخش داریم که می‌شود $836 = 288$ و برای مثال بعد از ۱۰ حرکت می‌شود $836 * 28 = 208$ و ... یعنی با جلو رفتن بازی این فاکتور کاهش می‌یابد.

سوال ۴ :

هرس کردن طوری انجام می‌شود که موارد مورد نیاز از بین نروند و برای مثلاً وقتی در یک خانه به یک مقدار مینیمم می‌رسیم ممکن است بتوانیم از قبل تشخیص دهیم که آیا در آن ارتفاع قرار است مقداری کمتر از مقدار فعلی بدست بیاوریم یا نه و اگر ببینیم نمی‌شود پس همه شاخه‌های زیرین این ارتفاع حذف شده و همین خانه انتخاب می‌شود پس تعداد شاخه‌های کمتری دیده می‌شوند و سرعت افزایش می‌یابد و چون دیتای آن خانه‌ها ارزشمند نبود دقت کم نمی‌شود.

سوال ۵ :

در مواجهه با حریف رندوم به این علت کارایی پایینی می‌آید که ما در این الگوریتم فرض می‌کنیم حریف بهترین حرکت‌ها را انتخاب می‌کند و بر اساس آن محاسبه می‌کنیم پس ممکن است خیلی از محاسبات اضافی باشند همینطور مکانیزم هرس کردن بر اساس انتخاب بهینه حریف برای مینیمایز کردن ما ساخته شده‌اند و در حالت رندوم چون انتخاب حریف معلوم نیست عملاً هرس کردن جواب نمی‌دهد و یا دقت خیلی پایینی دارد.