سوال ۱)

الف) حالت $N-1$

ب) فرض کنیم که می‌خواهیم در لیست L، مرتب‌سازی را به صورت صعودی انجام دهیم.
تابع هدف را در هر وضعیت، برابر تعداد اعدادی که از عدد سمت چپی خود کمتر هستند در نظر می‌گیریم.
تابع مقایسه (compare) ما بررسی می‌کند و اگر عدد سمت چپی بزرگ‌تر بود مقدار یک برمی‌گرداند و در غیر این صورت صفر برمی‌گرداند.

$$\sum_{i=0}^{N-2} compare(L[i], L[i+1])$$

تابع هدف ما نیز باید کمینه باشد، پس هر چه جمع این اعداد کمتر باشد ما به هدف خود که صفر کردن جمع این اعداد است، نزدیک‌تر شده‌ایم.

ج) بله، مثال‌هایی وجود دارد که ما به شانه (shoulder) می‌رسیم. مثلا در دنباله‌ی $4, 1, 2$ تابع هدف ما مقدار ۱ را نشان می‌دهد و حالت‌های همسایه هم برابر $1,4,2$ با هزینه‌ی ۱ و همچنین $4, 2, 1$ با مقدار ۲ هستند که هیچ‌کدام کمتر از حالت ابتدایی نیستند پس یعنی ما در مینیمم محلی قرار گرفته‌ایم.

د) تپه‌نوردی:

$$current \rightarrow [2, 7, 3, 9, 5] \qquad f = 2$$

$$current \rightarrow [2, 3,7, 9, 5] \qquad f = 1$$

$$current \rightarrow [2, 3,7, 5,9] \qquad f = 1$$

$$current \rightarrow [2, 3,5,7,9] \qquad f = 0$$

در بین همسایه‌ها بهترین را انتخاب کرده ام

سوال ۲)

الف) باید با سرعت بیشتری دما را کم کنیم زیرا باعث بزرگ‌تر شدن $\left|\frac{\Delta E}{T}\right|$ می‌شود و در نتیجه احتمال $e^{\frac{\Delta E}{T}}$ سریع‌تر کم می‌شود و به هم‌گرایی می‌رسیم.

ب) تپه‌نوردی، زیرا بدون اعمال تصادفی و به صورت سریع‌تر به سمت هدف حرکت می‌کند. مشکل این الگوریتم، گرفتار شدن در اکسترمم‌های محلی بود و می‌خواستیم با اعمال تصادفی این مشکل را حل کرده و اکسترمم کلی مسئله را پیدا کنیم که پیدا کردن آن برای ما هزینه‌ی اضافی داشت ولی اینجا با توجه به اینکه اکسترمم‌های محلی نداریم پس قطعا از تپه‌نوردی استفاده می‌کنیم.

ج) برای رهایی از این حالت باید از حرکات تصادفی کمک بگیریم و برای این کار باید احتمال $e^{\frac{\Delta E}{T}}$ را بالاتر ببریم پس دما را زیاد می‌کنیم.