a) A ≤p B e A é P.

Independente de P?=NP, não podemos afirmar nada sobre B. Ele pode pertencer a qualquer classe de problemas. O fato de existir uma redução em tempo polinomial para B não quer dizer que vamos conseguir **resolver** B em tempo polinomial.

b) A ≤p B e A é NP.

Independente de P?=NP, não podemos afirmar nada sobre B. Ele pode pertencer a qualquer classe de problemas. O fato de existir uma redução em tempo polinomial para B não quer dizer que vamos conseguir **verificar** B em tempo polinomial.

c) A ≤p B e A é NP-Completo.

Como podemos reduzir todo problema em NP em tempo polinomial para A e podemos reduzir A em tempo polinomial para B, por transitividade vemos que todo problema em NP pode ser reduzido em tempo polinomial para B, logo B é NP difícil.

Se P = NP, B é NP-difícil, podendo ou não pertencer a (P=NP-NPC).

Se P != NP, B pode ser tanto NP-difícil quanto NP-completo (se pertencer a NP).

d) A ≤p B e A é NP-Difícil.

A resposta é exatamente igual à questão anterior.

e) B ≤p A e A é P.

 Aqui, a questão P ?= NP é irrelevante. B está em P (veja a prova do teorema 7.31 de SIPSER). Note que é óbvio que, se dá para resolver A em tempo polinomial e A é pelo menos tão difícil quanto B, então conseguimos resolver B em tempo polinomial.

f) B ≤p A e A é NP.

 Se P = NP, a resposta se reduz à resposta anterior. Se P != NP, então B é pelo menos verificável em tempo polinomial (a prova é análoga ao teorema anterior, exceto que a máquina que resolve A em tempo polinomial é não determinística ao invés de determinística).

g) B ≤p A e A é NP-Completo.

 Independente de P ?= NP, a resposta é exatamente igual à anterior, porque o fato de A ser NP-difícil não acrescenta em nada na resposta. Acrescentaria se fosse o contrário, como nas letras “c” e “d”

h) B ≤p A e A é NP-Difícil.

 B pode pertencer a qualquer classe de linguagens e a explicação já está dada na letra “g”.