

ϵ -greedy

In [4]:

```
H = 10000 #nombre de patients
epsilon = 0.2

actions_Q = [0, 0, 0] #correspond aux Q_t(a)
actions_N = [0, 0, 0] #correspond aux N_t(a)

#actions cumulées : correspond aux N_t(a) en fonction du temps
actions_compteurs = [[0], [0], [0]]
```

In [5]:

```
for t in range(H):
    rand = np.random.random() #nombre aléatoire entre 0 et 1

    if rand < epsilon:
        a = np.random.randint(len(actions_q))
    else:
        a = indice_max(actions_Q)

    actions_N[a] += 1
    r = generer_reponse(actions_q, a) #on observe la réponse de l'action

    #on met à jour Q_t(a) (moyenne incrémentale)
    actions_Q[a] = actions_Q[a] + (1 / actions_N[a]) * (r - actions_Q[a])

    #on met à jour les compteurs de chaque action
    actions_compteurs[a].append(actions_compteurs[a][-1] + 1)
    for b in range(len(actions_q)):
        if b != a:
            actions_compteurs[b].append(actions_compteurs[b][-1])
```

In [6]:

```
actions_Q
```

Out[6]:

```
[0.9014133456904546, 0.11394302848575723, 0.8687589158345221]
```

In [7]:

```
plt.plot(actions_compteurs[0], label='0.9')
plt.plot(actions_compteurs[1], label='0.1')
plt.plot(actions_compteurs[2], label='0.88')

plt.xlabel('temps')
plt.ylabel('actions cumulées')
plt.legend()

plt.savefig('eps_greedy.png', dpi=300)
```

