Question 1

a) Last week, we computed the effective number of relativistic degrees of freedom $g_*(T)$ by counting the particles with mass $m_i < T$ at each T. This gave us the values of $g_*(T)$ at the plateaus shown in Fig. 1 below. However, $g_*(T)$ is a smooth function of temperature (except maybe at the QCD transition), not a series of sharp edges going down. If you were asked to compute this curve exactly at $T > 200$ MeV, how would you do it? Write down a formal expression for $g_*(T)$ valid at $T > 200$ MeV. You don't need to numerically evaluate your expression.

b) In Fig. 1 below, the curve splits into two different lines for $T \lesssim 0.5$ MeV (one solid, one dotted). This occurs because $g_*(T) \neq g_{*S}(T)$ at these low temperatures. Compute both $g_*(T)$ and $g_{*S}(T)$ after e^+e^- annihilation and confirm the values shown on the plot for $T \lesssim 0.05$ MeV. Which line shows $g_{*S}(T)$?

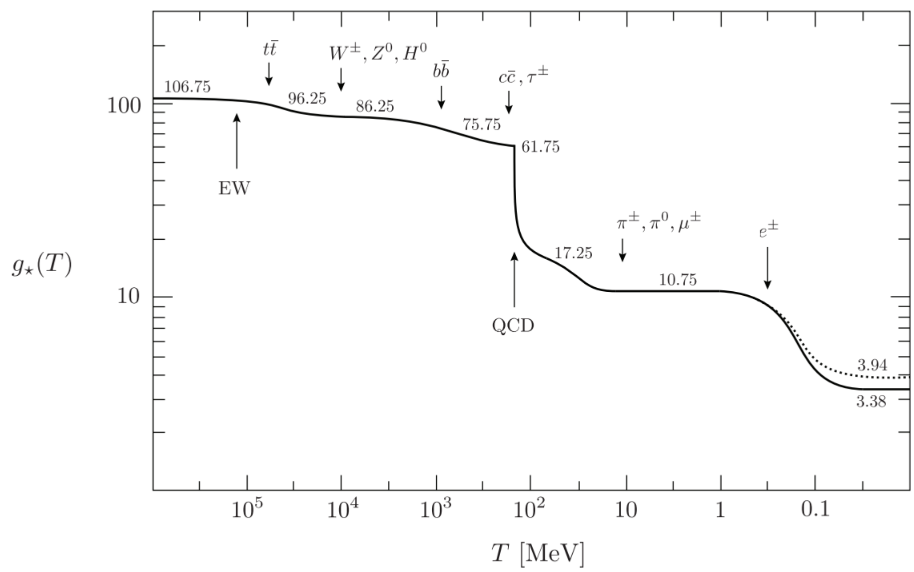


Fig. 1: Evolution of the effective number of relativistic degrees of freedom assuming the Standard Model particle content.

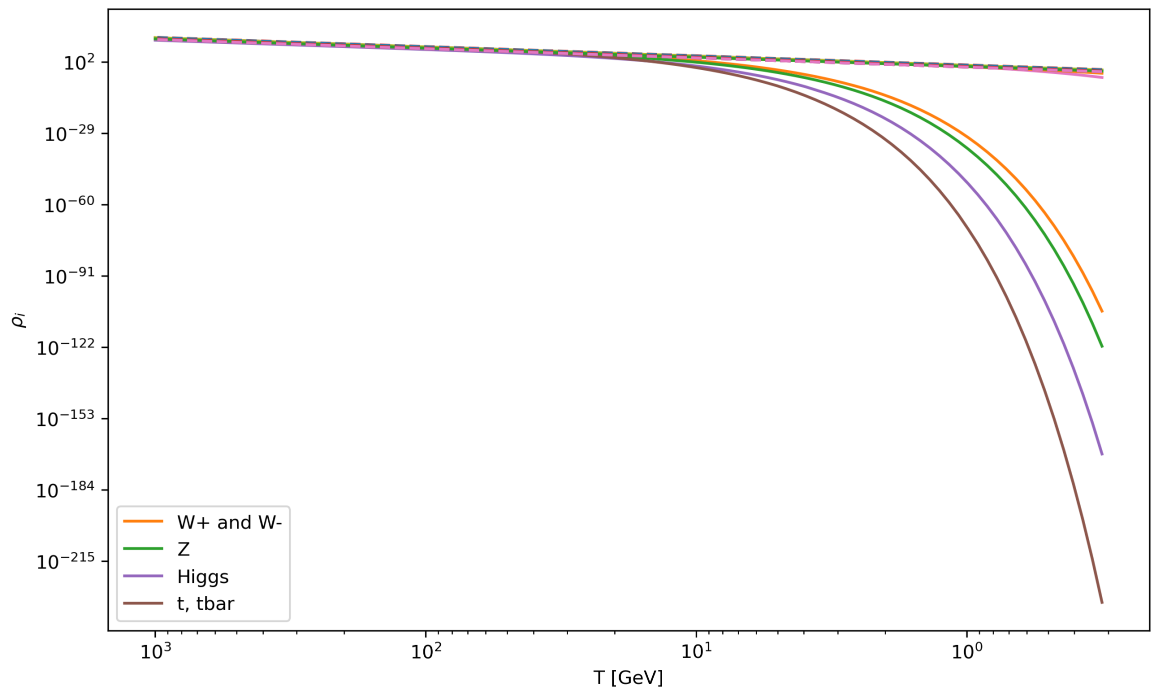
a) Start by recalling that g_* was defined as

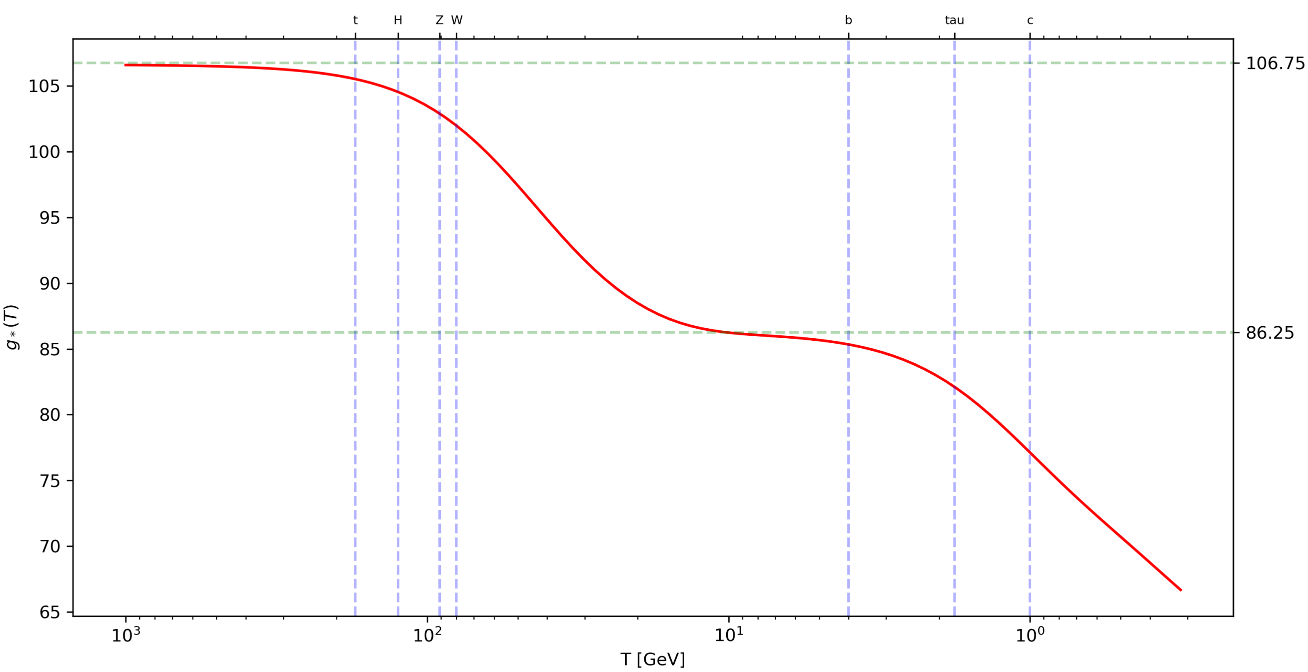
$$\rho(T) = \frac{\pi^2}{30} g_*(T) T^4 \tag{1}$$

So it is straightforward to solve for it:

$$\begin{aligned} g_*(T) &= \frac{30}{\pi^2 T^4} \rho(T) \\ &= \frac{30}{\pi^2 T^4} \sum_i \rho_i(T) \\ &= \frac{30}{\pi^2 T^4} \sum_i \frac{g_i}{2\pi^2} \int_0^\infty \frac{p^2 \sqrt{p^2 + m_i^2}}{e^{\sqrt{p^2+m_i^2}} \pm 1} dp \end{aligned} \tag{2}$$

Where the index i runs over all species, m_i is the mass of the ith species, + (−) for fermions (bosons), and g_i is the degrees of freedom of the ith species. To show that this works, we can go an extra step and implement this in python. Find below plots below and code at the end of problem (b).





b) After e^+e^- annihilation, all we have left are neutrinos and photons. As we know, at these (very) late times, the neutrino temperature is slightly lower than that of photons:

$$T_\nu = \left(\frac{4}{11}\right)^{1/3} T_\gamma \simeq 0.7137 T_\gamma \tag{3}$$

We can use this temperature to compute g_* and g_{*S} directly, again observing that the only species left are neutrinos and photons:

$$\begin{aligned} g_*(T_\gamma) &= \sum_{\text{bosons}} g_i \left(\frac{T_i}{T_\gamma}\right)^4 + \frac{7}{8} \sum_{\text{fermions}} g_i \left(\frac{T_i}{T}\right)^4 \\ &= 2 \cdot \left(\frac{T_\gamma}{T_\gamma}\right)^4 + \frac{7}{8} \cdot (2N_{\text{eff}}) \left(\frac{T_\nu}{T_\gamma}\right)^4 \\ &= 2 + \frac{14}{8} N_{\text{eff}} \left(\left[\frac{4}{11}\right]^{1/3}\right)^4 \\ &= 2 + \frac{14}{8}(3.044) \left(\frac{4}{11}\right)^{4/3} \\ &\approx 3.38 \end{aligned} \tag{4}$$

Similarly for g_{*S}:

$$\begin{aligned} g_{*S}(T_\gamma) &= \sum_{\text{bosons}} g_i \left(\frac{T_i}{T_\gamma}\right)^3 + \frac{7}{8} \sum_{\text{fermions}} g_i \left(\frac{T_i}{T}\right)^3 \\ &= 2 \cdot \left(\frac{T_\gamma}{T_\gamma}\right)^3 + \frac{7}{8} \cdot (2N_{\text{eff}}) \left(\frac{T_\nu}{T_\gamma}\right)^3 \\ &= 2 + \frac{14}{8} N_{\text{eff}} \left(\left[\frac{4}{11}\right]^{1/3}\right)^3 \\ &= 2 + \frac{14}{8}(3.044) \left(\frac{4}{11}\right) \\ &\approx 3.94 \end{aligned} \tag{5}$$

So g_* is the solid line and g_{*S} is the dotted line in Figure 1.

```python
import scipy.integrate as integrate
import numpy as np
import matplotlib.pyplot as plt

def integrand(p,T,mass,statistic):
    E = np.sqrt(p**2 + mass**2)
    return p**2 * E / (np.exp(E/T)+statistic)

def rho(T,species):
    mass = species[0]
    g = species[1]
    statistic = species[2]
    integral, error = integrate.quad(
        lambda pp: integrand(pp,T,mass,statistic),
        0,
        np.inf
        )
    return g*integral/(2*np.pi**2)


# Format:
# m_i, g_i, statistic. Mass in GeV
# Values taken from Baumann's Table 3.2
# Statistic is +1 for fermions, -1 for bosons
data = [
    [0,          2,  -1, "photons"],
    [80,         6,  -1, "W+ and W-"],
    [91,         3,  -1, "Z"],
    [0,          16, -1, "Gluons"],
    [125,        1,  1, "Higgs"],
    [173,        12, 1, "t, tbar"],
    [4,          12, 1, "b, bbar"],
    [1,          12, 1, "c, cbar"],
    [0.1,        12, 1, "s, sbar"],
    [0.005,      12, 1, "d, dbar"],
    [0.002,      12, 1, "u, ubar"],
    [1.777,      4,  1, "tau"],
    [0.106,      4,  1, "muon"],
    [0.000511,   4,  1, "electron"],
    [6.e-10,     2,  1, "tau neutrino"],
    [6.e-10,     2,  1, "mu neutrino"],
    [6.e-10,     2,  1, "e neutrino"]
    ]

temperatures = np.logspace(-0.5,3,100) # GeV

energy_densities = [[rho(T,species) for T in temperatures] for species in data]

plt.figure(figsize=(10,6))
for rho_species, data_row in zip(energy_densities,data):
    plt.plot(temperatures,
             rho_species,
             label=data_row[3] if data_row[0]>10 else None,
             linestyle="--" if data_row[0]<0.1 else "-"
             )
plt.xscale("log")
```

```
plt.yscale("log")
plt.xlabel("T [GeV]")
plt.ylabel(r"$\rho _i$")
plt.legend(loc="lower left")
plt.gca().invert_xaxis()
plt.savefig("densities.png",dpi=300,bbox_inches='tight')
plt.show()


g_star = 30*sum(np.array(energy_densities)) / (np.pi**2 * np.array(temperatures)
    **4)

# Double axes stuff
special_g = [86.25,106.75]
special_g_names = ["86.25","106.75"]

sepcial_T = [173,125,91,80,4,1.777,1]
sepcial_T_names = ["t","H","Z","W","b","tau","c"]

# Plot
fig, ax = plt.subplots(figsize=(12,6))
ax.plot(temperatures,g_star,color="red")
ax.set_xscale("log")
ax.set_xlabel("T [GeV]")
ax.set_ylabel(r"$g _* (T)$")
fig.gca().invert_xaxis()

secax = ax.secondary_yaxis('right')
secax.set_yticks(special_g)
secax.set_yticklabels(special_g_names)

secax = ax.secondary_xaxis('top')
secax.set_xticks(sepcial_T)
secax.set_xticklabels(sepcial_T_names,fontsize=7)
for value in sepcial_T:
    ax.axvline(x=value, color="blue",alpha=0.3,linestyle="--")
for value in special_g:
    ax.axhline(y=value, color="green",alpha=0.3,linestyle="--")

fig.savefig("g_star.png",dpi=300,bbox_inches='tight')
fig.show()
```